

# **The Book of Romance II**

---

A reference manual for Tootsville V, version 0.6.9.

**Bruce-Robert Pocock** <BRPocock@ciwta.org>

---

Copyright © 2008-2017 Bruce-Robert Pocock

Copyright © 2018-2021 The Corporation for Inter-World Tourism and Adventuring <https://ciwta.org/>

This manual is based upon manual-generating code taken from Declt 2.3.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided also that the section entitled “Copying” is included exactly as in the original.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be translated as well.

# Table of Contents

Copying .....	1
<b>1 Introduction .....</b>	<b>3</b>
1.1 Who are CIWTA? .....	3
1.2 What is Tootsville? .....	3
1.3 What is the Romance Game System? .....	3
1.4 Technology Stack .....	3
1.5 Affiliated Services .....	4
1.6 Clusters .....	4
1.7 Overview of Major Systems .....	4
1.7.1 Methods of Connecting .....	4
1.7.1.1 REST Requests .....	4
1.7.1.2 Infinity Mode communications .....	4
1.7.1.3 The Adult Sign-in Process .....	5
1.7.1.4 The Child Sign-in Process .....	5
1.7.1.5 The Server-to-Server Sign-In Process .....	5
1.7.2 In-Game Actions .....	6
1.7.2.1 Moving in the Game .....	6
1.7.2.2 Speech and Related Things .....	6
1.7.2.3 Game Events System (including Store Items) .....	6
1.7.2.4 Land Ownership .....	6
1.7.2.5 Clothing, Tools, and Equipment .....	7
1.7.2.6 Metronome .....	7
1.7.3 World Simulation .....	7
1.7.4 Server-to-Server Streams .....	7
1.7.5 The front-end .....	7
1.7.5.1 Coding Standard .....	7
1.7.5.2 Babylon.js .....	8
1.7.5.3 Gatekeeper .....	8
1.7.5.4 Peer-to-Peer Streams (WebRTC) .....	8
1.7.5.5 JSCL .....	8
1.8 Back Story .....	9
1.8.1 The Magic Mist and Mist Parrots .....	9
1.8.2 The Founding of Tootsville .....	9
1.8.3 The Classical Period .....	9
1.8.4 The Evil Mayor and Shade .....	9
1.8.5 The Destruction of Tootsville .....	9
1.8.6 The Revival .....	10
<b>2 Definitions .....</b>	<b>11</b>
<b>3 Package Choerogryllum .....</b>	<b>13</b>
3.1 Choerogryllum::Cal-Month .....	14

3.1.1	Function .....	14
3.1.2	File .....	14
3.2	Chœrogryllum::Cal-Month-Header .....	15
3.2.1	Function .....	15
3.2.2	File .....	15
3.3	Chœrogryllum::Cal-Month-Header.Html .....	16
3.3.1	Function .....	16
3.3.2	File .....	16
3.4	Chœrogryllum::Cal-Month.Html .....	17
3.4.1	Function .....	17
3.4.2	File .....	17
3.5	Chœrogryllum::Cal-Month/ Print-Holiday-Footnotes .....	18
3.5.1	Function .....	18
3.5.2	File .....	18
3.6	Chœrogryllum::Cal-Year .....	19
3.6.1	Function .....	19
3.6.2	File .....	19
3.7	Chœrogryllum::Date-String .....	20
3.7.1	Function .....	20
3.7.2	File .....	20
3.8	Chœrogryllum::Day-Of-Week* .....	21
3.8.1	Function .....	21
3.8.2	File .....	21
3.9	Chœrogryllum::Decode*-Universal-Time .....	22
3.9.1	Function .....	22
3.9.2	File .....	22
3.10	Chœrogryllum::Encode*-Universal-Time .....	23
3.10.1	Function .....	23
3.10.2	File .....	23
3.11	Chœrogryllum::Exponent-Digit .....	24
3.11.1	Function .....	24
3.11.2	File .....	24
3.12	Chœrogryllum::First-Weekday-Of-Month .....	25
3.12.1	Function .....	25
3.12.2	File .....	25
3.13	Chœrogryllum::Holiday-On .....	26
3.13.1	Function .....	26
3.13.2	Chœrogryllum Holiday .....	26
3.13.3	File .....	27
3.14	Chœrogryllum::Month* .....	28
3.14.1	Function .....	28
3.14.2	File .....	28
3.15	Chœrogryllum::This-Month .....	29
3.15.1	Function .....	29
3.15.2	File .....	29
3.16	Chœrogryllum::This-Year .....	30
3.16.1	Function .....	30
3.16.2	File .....	30

<b>4</b>	<b>Package Dreamhost</b> .....	<b>31</b>
4.1	Dreamhost:: <code>*Api-Key*</code> .....	32
4.1.1	Variable .....	32
4.2	Dreamhost:: <code>Cname-Already-On-Record</code> .....	33
4.2.1	Class .....	33
4.2.2	Slots .....	33
4.3	Dreamhost:: <code>Cname-Must-Be-Only-Record</code> .....	34
4.3.1	Class .....	34
4.3.2	Slots .....	34
4.4	Dreamhost:: <code>Dns-Add-Record</code> .....	35
4.4.1	Function .....	35
4.4.2	Result success .....	35
4.4.3	Possible Errors .....	35
4.4.4	File .....	35
4.5	Dreamhost:: <code>Dns-List-Records</code> .....	36
4.5.1	Function .....	36
4.5.2	File .....	36
4.6	Dreamhost:: <code>Dns-Remove-Record</code> .....	37
4.6.1	Function .....	37
4.6.2	File .....	37
4.7	Dreamhost:: <code>Dreamhost-Api-Error</code> .....	38
4.7.1	Class .....	38
4.7.2	Slots .....	38
4.8	Dreamhost:: <code>Dreamhost-Api-Error-With-Details</code> .....	39
4.8.1	Class .....	39
4.8.2	Slots .....	39
4.9	Dreamhost:: <code>Dreamhost-Api-Warning</code> .....	40
4.9.1	Class .....	40
4.9.2	Slots .....	40
4.10	Dreamhost:: <code>Dreamhost-Error-Details</code> .....	41
4.10.1	Function .....	41
4.11	Dreamhost:: <code>Internal-Error-Could-Not-Add-Record</code> .....	42
4.11.1	Class .....	42
4.11.2	Slots .....	42
4.12	Dreamhost:: <code>Internal-Error-Could-Not-Load-Zone</code> .....	43
4.12.1	Class .....	43
4.12.2	Slots .....	43
4.13	Dreamhost:: <code>Internal-Error-Updating-Zone</code> .....	44
4.13.1	Class .....	44
4.13.2	Slots .....	44
4.14	Dreamhost:: <code>Invalid-Record</code> .....	45
4.14.1	Class .....	45
4.14.2	Slots .....	45
4.15	Dreamhost:: <code>Invalid-Type</code> .....	46
4.15.1	Class .....	46
4.15.2	Slots .....	46
4.16	Dreamhost:: <code>Invalid-Value</code> .....	47
4.16.1	Class .....	47

4.16.2	Slots .....	47
4.17	Dreamhost::No-Record .....	48
4.17.1	Class .....	48
4.17.2	Slots .....	48
4.18	Dreamhost::No-Such-Zone .....	49
4.18.1	Class .....	49
4.18.2	Slots .....	49
4.19	Dreamhost::No-Type .....	50
4.19.1	Class .....	50
4.19.2	Slots .....	50
4.20	Dreamhost::No-Value .....	51
4.20.1	Class .....	51
4.20.2	Slots .....	51
4.21	Dreamhost::Record-Already-Exists-Not-Editable .....	52
4.21.1	Class .....	52
4.21.2	Slots .....	52
4.22	Dreamhost::Record-Already-Exists-Remove-First .....	53
4.22.1	Class .....	53
4.22.2	Slots .....	53
4.23	Dreamhost::Register-Dns-Name .....	54
4.23.1	Function .....	54
4.23.2	File .....	54
4.24	Dreamhost::Validate-Dns-Value .....	55
4.24.1	Function .....	55
4.24.2	File .....	55
<b>5</b>	<b>Package Rollbar .....</b>	<b>57</b>
5.1	Rollbar::*Access-Token* .....	58
5.1.1	Variable .....	58
5.2	Rollbar::*Code-Version* .....	59
5.2.1	Variable .....	59
5.3	Rollbar::*Environment* .....	60
5.3.1	Variable .....	60
5.4	Rollbar::*Framework* .....	61
5.4.1	Variable .....	61
5.5	Rollbar::*Person-Hook* .....	62
5.5.1	Variable .....	62
5.6	Rollbar::*Server* .....	63
5.6.1	Variable .....	63
5.7	Rollbar::*Valid-Notifier-Levels* .....	64
5.7.1	Variable .....	64
5.8	Rollbar::+Context-Forms+ .....	65
5.8.1	Variable .....	65
5.9	Rollbar::Backtrace-Frame-To-Plist .....	66
5.9.1	Function .....	66
5.9.2	File .....	66
5.10	Rollbar::Chain-Debugger-Hook .....	67
5.10.1	Function .....	67

5.10.2	File .....	67
5.11	Rollbar::Classify-Error-Level .....	68
5.11.1	Function .....	68
5.11.2	File .....	68
5.12	Rollbar::Condition-Telemetry .....	69
5.12.1	Function .....	69
5.12.2	File .....	69
5.13	Rollbar::Configure .....	70
5.13.1	Function .....	70
5.13.2	File .....	70
5.14	Rollbar::Constituent-Char-P .....	71
5.14.1	Function .....	71
5.14.2	File .....	71
5.15	Rollbar::Critical! .....	72
5.15.1	Function .....	72
5.15.2	File .....	72
5.16	Rollbar::Debug! .....	73
5.16.1	Function .....	73
5.16.2	File .....	73
5.17	Rollbar::Debugger-Hook .....	74
5.17.1	Function .....	74
5.17.2	File .....	74
5.18	Rollbar::Error! .....	75
5.18.1	Function .....	75
5.18.2	File .....	75
5.19	Rollbar::Escaped .....	76
5.19.1	Function .....	76
5.19.2	File .....	76
5.20	Rollbar::Find-Appropriate-Backtrace .....	77
5.20.1	Function .....	77
5.20.2	File .....	77
5.21	Rollbar::Format-Symbol-Name-Carefully .....	78
5.21.1	Function .....	78
5.21.2	File .....	78
5.22	Rollbar::Gather-Source-Info .....	79
5.22.1	Function .....	79
5.22.2	File .....	79
5.23	Rollbar::Http-Error .....	80
5.23.1	Class .....	80
5.23.2	Slots .....	80
5.24	Rollbar::Http-Error-Got-Uri .....	81
5.24.1	Function .....	81
5.25	Rollbar::Http-Error-Headers .....	82
5.25.1	Function .....	82
5.26	Rollbar::Http-Error-Status .....	83
5.26.1	Function .....	83
5.27	Rollbar::Http-Error-Status-Text .....	84
5.27.1	Function .....	84

5.28	Rollbar::Http-Error-Wanted-Uri .....	85
5.28.1	Function .....	85
5.29	Rollbar::Http-Successful-Request .....	86
5.29.1	Function .....	86
5.29.2	File .....	86
5.30	Rollbar::Info! .....	87
5.30.1	Function .....	87
5.30.2	File .....	87
5.31	Rollbar::Level-Is-Valid-P .....	88
5.31.1	Function .....	88
5.31.2	File .....	88
5.32	Rollbar::Make-Level-Notifier .....	89
5.32.1	Function .....	89
5.32.2	File .....	89
5.33	Rollbar::Notify .....	90
5.33.1	Function .....	90
5.33.2	File .....	90
5.34	Rollbar::Output-For-Level .....	91
5.34.1	Function .....	91
5.34.2	File .....	91
5.35	Rollbar::Package-Name-Can-Be-Unquoted-P .....	92
5.35.1	Function .....	92
5.35.2	File .....	92
5.36	Rollbar::Pretty-Function-Name .....	93
5.36.1	Function .....	93
5.36.2	File .....	93
5.37	Rollbar::Pretty-Symbol-Name .....	94
5.37.1	Function .....	94
5.37.2	File .....	94
5.38	Rollbar::Quoted .....	95
5.38.1	Function .....	95
5.38.2	File .....	95
5.39	Rollbar::Redact-Directory .....	96
5.39.1	Function .....	96
5.39.2	File .....	96
5.40	Rollbar::Report-Server-Info .....	97
5.40.1	Function .....	97
5.40.2	File .....	97
5.41	Rollbar::Report-Telemetry .....	98
5.41.1	Function .....	98
5.41.2	File .....	98
5.42	Rollbar::Request-Telemetry .....	99
5.42.1	Function .....	99
5.42.2	File .....	99
5.43	Rollbar::Rollbar-Notify-Deployment .....	100
5.43.1	Function .....	100
5.43.2	File .....	100
5.44	Rollbar::Sanitize-File-Name .....	101



5.44.1	Function	101
5.44.2	File	101
5.45	Rollbar::Send-Rollbar-Notification	102
5.45.1	Function	102
5.45.2	File	102
5.46	Rollbar::Symbol-Is-Exported-P	103
5.46.1	Function	103
5.46.2	File	103
5.47	Rollbar::Symbol-Name-Can-Be-Unquoted-P	104
5.47.1	Function	104
5.47.2	File	104
5.48	Rollbar::Warning!	105
5.48.1	Function	105
5.48.2	File	105
5.49	Rollbar::With-Configuration	106
5.49.1	Macro	106
5.49.2	File	106
5.50	Rollbar::With-Rollbar-For-Debugger	107
5.50.1	Macro	107
5.50.2	File	107
<b>6</b>	<b>Package Thread-Pool-Taskmaster</b>	<b>109</b>
6.1	Thread-Pool-Taskmaster::*Developmentp*	110
6.1.1	Variable	110
6.2	Thread-Pool-Taskmaster::*Mulligans*	111
6.2.1	Variable	111
6.3	Thread-Pool-Taskmaster::+Max-Queue-Size-For-Thread-Pool+	112
6.3.1	Variable	112
6.4	Thread-Pool-Taskmaster::+Single-Core-Threads+	113
6.4.1	Variable	113
6.5	Thread-Pool-Taskmaster::+Threads-Per-Core+	114
6.5.1	Variable	114
6.6	Thread-Pool-Taskmaster::Cores*Threads-Per-Core	115
6.6.1	Function	115
6.6.2	File	115
6.7	Thread-Pool-Taskmaster::Make-Thread-Name	116
6.7.1	Function	116
6.7.2	File	116
6.8	Thread-Pool-Taskmaster::Name-Idle-Threads-Sequentially	117
6.8.1	Function	117
6.8.2	File	117
6.9	Thread-Pool-Taskmaster::Named-Thread-Pool-Runner	118
6.9.1	Macro	118
6.9.2	File	118
6.10	Thread-Pool-Taskmaster::Safe-Client-As-String	119
6.10.1	Function	119
6.10.2	File	119
6.11	Thread-Pool-Taskmaster::Swank-Connected-P	120

6.11.1	Function .....	120
6.11.2	File .....	120
6.12	Thread-Pool-Taskmaster::Taskmaster-Thread-Pool .....	121
6.12.1	Function .....	121
6.12.2	SetF Function .....	121
6.13	Thread-Pool-Taskmaster::Taskmaster-Thread-Pool-Channel ..	122
6.13.1	Function .....	122
6.13.2	SetF Function .....	122
6.14	Thread-Pool-Taskmaster::Thread-Pool-Taskmaster .....	123
6.14.1	Class .....	123
6.14.2	Slots .....	123
6.15	Thread-Pool-Taskmaster::With-Mulligan-Handlers .....	124
6.15.1	Macro .....	124
6.15.2	File .....	124
6.16	Thread-Pool-Taskmaster::With-Pool-Thread-Restarts .....	125
6.16.1	Macro .....	125
6.16.2	File .....	125
<b>7</b>	<b>Package Tootsville-User .....</b>	<b>127</b>
7.1	Tootsville-User::\$ .....	128
7.1.1	Function .....	128
7.1.2	Usage .....	128
7.1.3	Example .....	128
7.1.4	File .....	128
7.2	Tootsville-User::*Apropos .....	129
7.2.1	Function .....	129
7.2.2	Usage .....	129
7.2.3	Example .....	129
7.2.4	File .....	129
7.3	Tootsville-User::*Time .....	130
7.3.1	Function .....	130
7.3.2	Usage .....	130
7.3.3	Example .....	130
7.3.4	Example Reply .....	130
7.3.5	Changes from 1.2 to 2.0 .....	130
7.3.6	File .....	130
7.4	Tootsville-User::*Warn .....	131
7.4.1	Function .....	131
7.4.2	Usage .....	131
7.4.3	Examples .....	131
7.4.4	Reason Codes .....	131
7.4.5	Changes from 1.2 to 2.0 .....	131
7.4.6	File .....	131
7.5	Tootsville-User::Addevent .....	132
7.5.1	Function .....	132
7.5.2	Usage .....	132
7.5.3	Examples .....	132
7.5.4	File .....	132

7.6	Tootsville-User::Agent	133
7.6.1	Function	133
7.6.2	Usage	133
7.6.3	File	133
7.7	Tootsville-User::Askme	134
7.7.1	Function	134
7.7.2	Usage	134
7.7.3	200 OK	134
7.7.4	File	134
7.8	Tootsville-User::At	135
7.8.1	Function	135
7.8.2	Usage	135
7.8.3	Examples	135
7.8.4	File	135
7.9	Tootsville-User::Ban	136
7.9.1	Function	136
7.9.2	Usage	136
7.9.3	Examples	136
7.9.4	File	136
7.10	Tootsville-User::Banhammer	137
7.10.1	Function	137
7.10.2	Usage	137
7.10.3	Parameters	137
7.10.4	File	137
7.11	Tootsville-User::Beam	138
7.11.1	Function	138
7.11.2	Usage	138
7.11.3	Changes from 1.2 to 2.0	138
7.11.4	File	138
7.12	Tootsville-User::Census	139
7.12.1	Function	139
7.12.2	Usage	139
7.12.3	Examples	139
7.12.4	File	139
7.13	Tootsville-User::Clearbadge	140
7.13.1	Function	140
7.13.2	Usage	140
7.13.3	Examples	140
7.13.4	Badges	140
7.13.5	Spots	140
7.13.6	File	140
7.14	Tootsville-User::Clearcache	141
7.14.1	Function	141
7.14.2	Usage	141
7.14.3	Example	141
7.14.4	File	141
7.15	Tootsville-User::Clearevent	142
7.15.1	Function	142

7.15.2	Usage .....	142
7.15.3	Examples .....	142
7.15.4	Changes from 1.2 to 2.0 .....	142
7.15.5	File .....	142
7.16	Tootsville-User::Clearvar .....	143
7.16.1	Function .....	143
7.16.2	Usage .....	143
7.16.3	Examples .....	143
7.16.4	410 Gone.....	143
7.16.5	File .....	143
7.17	Tootsville-User::Clonerroom .....	144
7.17.1	Function .....	144
7.17.2	Usage .....	144
7.17.3	Legacy Operator Command .....	144
7.17.4	File .....	144
7.18	Tootsville-User::Createroom .....	145
7.18.1	Function .....	145
7.18.2	Usage .....	145
7.18.3	Example .....	145
7.18.4	Legacy Operator Command .....	145
7.18.5	File .....	145
7.19	Tootsville-User::Dbcpinfo.....	146
7.19.1	Function .....	146
7.19.2	Usage .....	146
7.19.3	File .....	146
7.20	Tootsville-User::Doc .....	147
7.20.1	Function .....	147
7.20.2	Usage .....	147
7.20.3	Examples .....	147
7.20.4	File .....	147
7.21	Tootsville-User::Doodle.....	148
7.21.1	Function .....	148
7.21.2	Usage .....	148
7.21.3	Examples .....	148
7.21.4	File .....	148
7.22	Tootsville-User::Doodle-Pattern .....	149
7.22.1	Function .....	149
7.22.2	Usage .....	149
7.22.3	Example .....	149
7.22.4	File .....	149
7.23	Tootsville-User::Dress .....	150
7.23.1	Function .....	150
7.23.2	Usage .....	150
7.23.3	Examples .....	150
7.23.4	File .....	150
7.24	Tootsville-User::Drop.....	151
7.24.1	Function .....	151
7.24.2	File .....	151

7.25	Tootsville-User::Dropkick	152
7.25.1	Function	152
7.25.2	Usage	152
7.25.3	Example	152
7.25.4	File	152
7.26	Tootsville-User::Dumpthreads	153
7.26.1	Function	153
7.26.2	Usage	153
7.26.3	Example	153
7.26.4	File	153
7.27	Tootsville-User::Enablepathfinder	154
7.27.1	Function	154
7.27.2	Usage	154
7.27.3	Examples	154
7.27.4	File	154
7.28	Tootsville-User::Evacuate	155
7.28.1	Function	155
7.28.2	Usage	155
7.28.3	Example	155
7.28.4	File	155
7.29	Tootsville-User::Filter	156
7.29.1	Function	156
7.29.2	Usage	156
7.29.3	File	156
7.30	Tootsville-User::Finger	157
7.30.1	Function	157
7.30.2	Usage	157
7.30.3	Examples	157
7.30.4	Changes from 1.2 to 2.0	157
7.30.5	Response	157
7.30.6	File	157
7.31	Tootsville-User::Flush	158
7.31.1	Function	158
7.31.2	Usage	158
7.31.3	Changes from 1.2 to 2.0	158
7.31.4	File	158
7.32	Tootsville-User::Game	159
7.32.1	Function	159
7.32.2	Usage	159
7.32.3	File	159
7.33	Tootsville-User::Gc	160
7.33.1	Function	160
7.33.2	Usage	160
7.33.3	File	160
7.34	Tootsville-User::Getconfig	161
7.34.1	Function	161
7.34.2	Usage	161
7.34.3	Example	161

7.34.4	Changes from 1.2 to 2.0 .....	161
7.34.5	File .....	161
7.35	Tootsville-User::Getevents .....	162
7.35.1	Function .....	162
7.35.2	Usage .....	162
7.35.3	Example .....	162
7.35.4	See Also.....	162
7.35.5	File .....	162
7.36	Tootsville-User::Getmotd .....	163
7.36.1	Function .....	163
7.36.2	Usage .....	163
7.36.3	Example .....	163
7.36.4	File .....	163
7.37	Tootsville-User::Getschedule .....	164
7.37.1	Function .....	164
7.37.2	File .....	164
7.38	Tootsville-User::Getschedulefor .....	165
7.38.1	Function .....	165
7.38.2	File .....	165
7.39	Tootsville-User::Getuvar .....	166
7.39.1	Function .....	166
7.39.2	Usage .....	166
7.39.3	Examples .....	166
7.39.4	See Also.....	166
7.39.5	File .....	166
7.40	Tootsville-User::Getuvars .....	167
7.40.1	Function .....	167
7.40.2	Usage .....	167
7.40.3	Examples .....	167
7.40.4	See Also.....	167
7.40.5	File .....	167
7.41	Tootsville-User::Getvar .....	168
7.41.1	Function .....	168
7.41.2	Usage .....	168
7.41.3	Examples .....	168
7.41.4	See Also.....	168
7.41.5	File .....	168
7.42	Tootsville-User::Getvars .....	169
7.42.1	Function .....	169
7.42.2	Usage .....	169
7.42.3	Examples .....	169
7.42.4	See Also.....	169
7.42.5	File .....	169
7.43	Tootsville-User::Git-Pull.....	170
7.43.1	Function .....	170
7.43.2	Usage .....	170
7.43.3	Effects .....	170
7.43.4	File .....	170

7.44	Tootsville-User::Give	171
7.44.1	Function	171
7.44.2	Usage	171
7.44.3	Example	171
7.44.4	File	171
7.45	Tootsville-User::Givehead	172
7.45.1	Function	172
7.45.2	Usage	172
7.45.3	Example	172
7.45.4	File	172
7.46	Tootsville-User::Grant	173
7.46.1	Function	173
7.46.2	Usage	173
7.46.3	Example	173
7.46.4	File	173
7.47	Tootsville-User::Headcount	174
7.47.1	Function	174
7.47.2	Usage	174
7.47.3	Examples	174
7.47.4	Headcount All	174
7.47.5	Headcount Members	174
7.47.6	Headcount Room	174
7.47.7	Headcount Highwater	174
7.47.8	File	174
7.48	Tootsville-User::Infinity-Stats	175
7.48.1	Function	175
7.48.2	Usage	175
7.48.3	Example	175
7.48.4	File	175
7.49	Tootsville-User::Inv	176
7.49.1	Function	176
7.49.2	Usage	176
7.49.3	File	176
7.50	Tootsville-User::Kick	177
7.50.1	Function	177
7.50.2	Usage	177
7.50.3	Example	177
7.50.4	Reason Codes	177
7.50.5	Reason Codes from 1.2	177
7.50.6	File	178
7.51	Tootsville-User::King	179
7.51.1	Function	179
7.51.2	Usage	179
7.51.3	Example	179
7.51.4	File	179
7.52	Tootsville-User::Liftban	180
7.52.1	Function	180
7.52.2	Usage	180

7.52.3	Example .....	180
7.52.4	File .....	180
7.53	Tootsville-User::Loadlists .....	181
7.53.1	Function .....	181
7.53.2	Usage .....	181
7.53.3	File .....	181
7.54	Tootsville-User::Mem .....	182
7.54.1	Function .....	182
7.54.2	Usage .....	182
7.54.3	Example .....	182
7.54.4	Example report .....	182
7.54.5	Changes from 1.2 to 2.0 .....	182
7.54.6	File .....	182
7.55	Tootsville-User::Metronome .....	183
7.55.1	Function .....	183
7.55.2	Usage .....	183
7.55.3	Examples .....	183
7.55.3.1	Options .....	183
7.55.4	Changes from 1.2 to 2.0 .....	183
7.55.5	File .....	183
7.56	Tootsville-User::Motd .....	184
7.56.1	Function .....	184
7.56.2	Usage .....	184
7.56.3	Example .....	184
7.56.4	Changes from 1.2 to 2.0 .....	184
7.56.5	File .....	184
7.57	Tootsville-User::Mute .....	185
7.57.1	Function .....	185
7.57.2	See also .....	185
7.57.3	File .....	185
7.58	Tootsville-User::Nuke .....	186
7.58.1	Function .....	186
7.58.2	Usage .....	186
7.58.3	Example .....	186
7.58.4	Results .....	186
7.58.5	Rationale .....	186
7.58.6	Rationale for version 1.2 .....	186
7.58.7	File .....	186
7.59	Tootsville-User::Parentapproves .....	187
7.59.1	Function .....	187
7.59.2	Usage .....	187
7.59.3	Example .....	187
7.59.4	Limitations .....	187
7.59.5	File .....	187
7.60	Tootsville-User::Ping .....	188
7.60.1	Function .....	188
7.60.2	Usage .....	188
7.60.3	Example .....	188



7.60.4	Reply .....	188
7.60.5	File .....	188
7.61	Tootsville-User::Place .....	189
7.61.1	Function .....	189
7.61.2	Usage .....	189
7.61.3	Examples .....	190
7.61.4	#download Placing a download trigger item .....	190
7.61.5	#exit Placing a transwarp conduit .....	190
7.61.5.1	Changes from 1.2 to 2.0 .....	190
7.61.6	#fountain Placing a magic fountain .....	190
7.61.7	item Placing an item .....	190
7.61.7.1	Changes from 1.2 to 2.0 .....	191
7.61.8	#item2.....	191
7.61.8.1	Changes from 1.2 to 2.0 .....	191
7.61.9	#place Placing a Place designator.....	191
7.61.10	room Placing a “room” (spot) marker.....	191
7.61.11	#shop Placing a shop item.....	191
7.61.12	#mini Placing a minigame.....	191
7.61.12.1	Changes from 1.2 to 2.0 .....	191
7.61.13	#snowball Placing a snowball source pile.....	191
7.61.14	#unwalk Placing an unwalkable space.....	191
7.61.15	vitem Placing an item-gifting item .....	192
7.61.15.1	Changes from 1.2 to 2.0 .....	192
7.61.16	#walk Placing a walkable space.....	192
7.61.17	Implementation note .....	192
7.61.18	File .....	192
7.62	Tootsville-User::Purgephysics.....	193
7.62.1	Function .....	193
7.62.2	Changes from 1.2 to 2.0 .....	193
7.62.3	File .....	193
7.63	Tootsville-User::Push-Script .....	194
7.63.1	Function .....	194
7.63.2	Usage .....	194
7.63.3	Example .....	194
7.63.4	File .....	194
7.64	Tootsville-User::Quick-Reload .....	195
7.64.1	Function .....	195
7.64.2	Usage .....	195
7.64.3	Effects .....	195
7.64.4	File .....	195
7.65	Tootsville-User::Rc .....	196
7.65.1	Function .....	196
7.65.2	Usage .....	196
7.65.3	Example .....	196
7.65.4	File .....	196
7.66	Tootsville-User::Reboot.....	197
7.66.1	Function .....	197
7.66.2	Usage .....	197

7.66.3	Example .....	197
7.66.4	Actual Effects .....	197
7.66.5	File .....	197
7.67	Tootsville-User::Reloadconfig .....	198
7.67.1	Function .....	198
7.67.2	Usage .....	198
7.67.3	Example .....	198
7.67.4	Effect .....	198
7.67.5	File .....	198
7.68	Tootsville-User::Retire .....	199
7.68.1	Function .....	199
7.68.2	Usage .....	199
7.68.3	Examples .....	199
7.68.4	File .....	199
7.69	Tootsville-User::Run .....	200
7.69.1	Function .....	200
7.69.2	USave .....	200
7.69.3	Examples .....	200
7.69.4	Changes from 1.2 to 2.0 .....	200
7.69.5	File .....	200
7.70	Tootsville-User::Saveroomvars .....	201
7.70.1	Function .....	201
7.70.2	Legacy Usage (1.2) .....	201
7.70.3	File .....	201
7.71	Tootsville-User::Scotty .....	202
7.71.1	Function .....	202
7.71.2	Usage .....	202
7.71.3	Examples .....	202
7.71.4	Changes from 1.2 to 2.0 .....	202
7.71.5	File .....	202
7.72	Tootsville-User::Script .....	203
7.72.1	Function .....	203
7.72.2	Usage .....	203
7.72.3	Example .....	203
7.72.4	File .....	203
7.73	Tootsville-User::Server-List .....	204
7.73.1	Function .....	204
7.73.2	Usage .....	204
7.73.3	Example .....	204
7.73.4	File .....	204
7.74	Tootsville-User::Setavatarcolors .....	205
7.74.1	Function .....	205
7.74.2	Usage .....	205
7.74.3	Examples .....	205
7.74.4	File .....	205
7.75	Tootsville-User::Setbadge .....	206
7.75.1	Function .....	206
7.75.2	Usage .....	206

7.75.3	Examples .....	206
7.75.4	File .....	206
7.76	Tootsville-User::Setconfig .....	207
7.76.1	Function .....	207
7.76.2	Usage .....	207
7.76.3	Example .....	207
7.76.4	File .....	207
7.77	Tootsville-User::Setstafflevel .....	208
7.77.1	Function .....	208
7.77.2	File .....	208
7.78	Tootsville-User::Setuvar .....	209
7.78.1	Function .....	209
7.78.2	Usage .....	209
7.78.3	Example .....	209
7.78.4	File .....	209
7.79	Tootsville-User::Setvar .....	210
7.79.1	Function .....	210
7.79.2	Description from Romance 1.2 .....	210
7.79.3	Examples .....	210
7.79.4	File .....	210
7.80	Tootsville-User::Shanghai .....	211
7.80.1	Function .....	211
7.80.2	File .....	211
7.81	Tootsville-User::Shout .....	212
7.81.1	Function .....	212
7.81.2	Usage .....	212
7.81.3	Examples .....	212
7.81.4	File .....	212
7.82	Tootsville-User::Spawnroom .....	213
7.82.1	Function .....	213
7.82.2	Usage .....	213
7.82.3	Changes from 1.2 to 2.0 .....	213
7.82.4	Legacy 1.2 Documentation .....	213
7.82.5	Usage in 1.2 .....	213
7.82.6	Examples of 1.2 syntax .....	213
7.82.7	File .....	213
7.83	Tootsville-User::Spawnzone .....	214
7.83.1	Function .....	214
7.83.2	Usage .....	214
7.83.3	Examples .....	214
7.83.4	File .....	214
7.84	Tootsville-User::Speak .....	215
7.84.1	Function .....	215
7.84.2	Usage .....	215
7.84.3	Examples .....	215
7.84.4	File .....	215
7.85	Tootsville-User::Stfu .....	216
7.85.1	Function .....	216

7.85.2	Usage .....	216
7.85.3	Example .....	216
7.85.4	Effects .....	216
7.85.5	File .....	216
7.86	Tootsville-User::Testcensor .....	217
7.86.1	Function .....	217
7.86.2	Usage .....	217
7.86.3	Examples .....	217
7.86.4	File .....	217
7.87	Tootsville-User::Unbuild .....	218
7.87.1	Function .....	218
7.87.2	Usage .....	218
7.87.3	Example .....	218
7.87.4	Changes from 1.2 to 2.0 .....	218
7.87.5	File .....	218
7.88	Tootsville-User::Uptime .....	219
7.88.1	Function .....	219
7.88.2	File .....	219
7.89	Tootsville-User::V .....	220
7.89.1	Function .....	220
7.89.2	Usage .....	220
7.89.3	Example .....	220
7.89.4	See also .....	220
7.89.5	Changes from 1.2 to 2.0 .....	220
7.89.6	File .....	220
7.90	Tootsville-User::Verbosebugs .....	221
7.90.1	Function .....	221
7.90.2	Usage .....	221
7.90.3	Impact .....	221
7.90.4	File .....	221
7.91	Tootsville-User::Wall .....	222
7.91.1	Function .....	222
7.91.2	Usage .....	222
7.91.3	Example .....	222
7.91.4	File .....	222
7.92	Tootsville-User::Wallops .....	223
7.92.1	Function .....	223
7.92.2	Usage .....	223
7.92.3	Exampleyy .....	223
7.92.4	File .....	223
7.93	Tootsville-User::Wallzones .....	224
7.93.1	Function .....	224
7.93.2	Usage .....	224
7.93.3	Example .....	224
7.93.4	Changes from 1.2 to 2.0 .....	224
7.93.5	File .....	224
7.94	Tootsville-User::Whatis .....	225
7.94.1	Function .....	225

7.94.2	Usage .....	225
7.94.3	Example .....	225
7.94.4	File .....	225
7.95	Tootsville-User::Whereami .....	226
7.95.1	Function .....	226
7.95.2	Usage .....	226
7.95.3	Example .....	226
7.95.4	File .....	226
7.96	Tootsville-User::Whereis .....	227
7.96.1	Function .....	227
7.96.2	Usage .....	227
7.96.3	Examples .....	227
7.96.4	File .....	227
7.97	Tootsville-User::Who .....	228
7.97.1	Function .....	228
7.97.2	Usage .....	228
7.97.3	Examples .....	228
7.97.4	File .....	228
7.98	Tootsville-User::Whoami .....	229
7.98.1	Function .....	229
7.98.2	Usage .....	229
7.98.3	Example .....	229
7.98.4	File .....	229
7.99	Tootsville-User::Whoareyou .....	230
7.99.1	Function .....	230
7.99.2	Usage .....	230
7.99.3	Example .....	230
7.99.4	Example Response .....	230
7.99.5	Changes from 1.2 to 2.0 .....	230
7.99.6	File .....	230
7.100	Tootsville-User::Ws-Bandwidth-By-Source .....	231
7.100.1	Function .....	231
7.100.2	Usage .....	231
7.100.3	Example .....	231
7.100.4	File .....	231
7.101	Tootsville-User::Ws-Stats .....	232
7.101.1	Function .....	232
7.101.2	Usage .....	232
7.101.3	Example .....	232
7.101.4	File .....	232
7.102	Tootsville-User::Zoom .....	233
7.102.1	Function .....	233
7.102.2	Changes from 1.2 to 2.0 .....	233
7.102.3	File .....	233

<b>8</b>	<b>Package Tootsville</b> .....	<b>235</b>
8.1	Tootsville::%Operator-Place-Download .....	236
8.1.1	Function .....	236
8.1.2	File .....	236
8.2	Tootsville::%Operator-Place-Exit .....	237
8.2.1	Function .....	237
8.2.2	File .....	237
8.3	Tootsville::%Operator-Place-Fountain .....	238
8.3.1	Function .....	238
8.3.2	File .....	238
8.4	Tootsville::%Operator-Place-Game .....	239
8.4.1	Function .....	239
8.4.2	File .....	239
8.5	Tootsville::%Operator-Place-Item .....	240
8.5.1	Function .....	240
8.5.2	File .....	240
8.6	Tootsville::%Operator-Place-Mini .....	241
8.6.1	Function .....	241
8.6.2	File .....	241
8.7	Tootsville::%Operator-Place-Place .....	242
8.7.1	Function .....	242
8.7.2	File .....	242
8.8	Tootsville::%Operator-Place-Room .....	243
8.8.1	Function .....	243
8.8.2	File .....	243
8.9	Tootsville::%Operator-Place-Shop .....	244
8.9.1	Function .....	244
8.9.2	File .....	244
8.10	Tootsville::%Operator-Place-Snowball .....	245
8.10.1	Function .....	245
8.10.2	File .....	245
8.11	Tootsville::%Operator-Place-Unwalk .....	246
8.11.1	Function .....	246
8.11.2	File .....	246
8.12	Tootsville::%Operator-Place-Vitem .....	247
8.12.1	Function .....	247
8.12.2	File .....	247
8.13	Tootsville::%Operator-Place-Walk .....	248
8.13.1	Function .....	248
8.13.2	File .....	248
8.14	Tootsville::%Parse-Operator-Place-Where .....	249
8.14.1	Function .....	249
8.14.2	File .....	249
8.15	Tootsville::*403.Json-Bytes* .....	250
8.15.1	Variable .....	250
8.16	Tootsville::*Acceptors* .....	251
8.16.1	Variable .....	251
8.17	Tootsville::*Application-Root* .....	252

8.17.1	Variable	252
8.18	Tootsville:*Async-Channel*	253
8.18.1	Variable	253
8.19	Tootsville:*Async-Tasks*	254
8.19.1	Variable	254
8.20	Tootsville:*Banhammer*	255
8.20.1	Variable	255
8.21	Tootsville:*Build-Date*	256
8.21.1	Variable	256
8.22	Tootsville:*Cassandra-Blacklist*	257
8.22.1	Variable	257
8.23	Tootsville:*Cassandra-Redlist*	258
8.23.1	Variable	258
8.24	Tootsville:*Client*	259
8.24.1	Variable	259
8.25	Tootsville:*Cluster*	260
8.25.1	Variable	260
8.26	Tootsville:*Compilation*	261
8.26.1	Variable	261
8.27	Tootsville:*Compiled*	262
8.27.1	Variable	262
8.28	Tootsville:*Config-File*	263
8.28.1	Variable	263
8.29	Tootsville:*Db*	264
8.29.1	Variable	264
8.30	Tootsville:*Dbi-Connection*	265
8.30.1	Variable	265
8.31	Tootsville:*Elevation-Map*	266
8.31.1	Variable	266
8.32	Tootsville:*Endpoint-List*	267
8.32.1	Variable	267
8.33	Tootsville:*Endpoints*	268
8.33.1	Variable	268
8.34	Tootsville:*Extensions-For-Content-Types*	287
8.34.1	Variable	287
8.35	Tootsville:*Google-Account-Keys-Refresh*	288
8.35.1	Variable	288
8.36	Tootsville:*Habitat-Map*	289
8.36.1	Variable	289
8.37	Tootsville:*Http-Status-Message*	290
8.37.1	Variable	290
8.38	Tootsville:*Humidity-Field*	292
8.38.1	Variable	292
8.39	Tootsville:*Ice-Credentials*	293
8.39.1	Variable	293
8.40	Tootsville:*Infinity-Ops*	294
8.40.1	Variable	294
8.41	Tootsville:*Infinity-Rest-Requests*	295

8.41.1	Variable	295
8.42	Tootsville:*Infinity-Stream-Requests*	296
8.42.1	Variable	296
8.43	Tootsville:*Infinity-Users*	297
8.43.1	Variable	297
8.44	Tootsville:*Infinity-Websocket-Resource*	298
8.44.1	Variable	298
8.45	Tootsville:*Maintenance-Tasks-Performed*	299
8.45.1	Variable	299
8.46	Tootsville:*Metronome*	300
8.46.1	Variable	300
8.47	Tootsville:*Metronome-Next-Tick*	301
8.47.1	Variable	301
8.48	Tootsville:*Metronome-Run*	302
8.48.1	Variable	302
8.49	Tootsville:*Metronome-Task-Lock*	303
8.49.1	Variable	303
8.50	Tootsville:*Metronome-Tasks*	304
8.50.1	Variable	304
8.51	Tootsville:*Motd*	305
8.51.1	Variable	305
8.52	Tootsville:*Npc-List*	306
8.52.1	Variable	306
8.53	Tootsville:*Post-Tests-Queue*	307
8.53.1	Variable	307
8.54	Tootsville:*Robots*	308
8.54.1	Variable	308
8.55	Tootsville:*Running-Main-Loop*	309
8.55.1	Variable	309
8.56	Tootsville:*Started*	310
8.56.1	Variable	310
8.57	Tootsville:*Tcp-Clients*	311
8.57.1	Variable	311
8.58	Tootsville:*Tcp-Listener*	312
8.58.1	Variable	312
8.59	Tootsville:*Tcp-Peer-Traffic*	313
8.59.1	Variable	313
8.60	Tootsville:*The-Metronome-Thread*	314
8.60.1	Variable	314
8.61	Tootsville:*Toot*	315
8.61.1	Variable	315
8.62	Tootsville:*Trace-Output-Heartbeat-Time*	316
8.62.1	Variable	316
8.63	Tootsville:*User*	317
8.63.1	Variable	317
8.64	Tootsville:*Utc-Timezone*	318
8.64.1	Variable	318
8.65	Tootsville:*Verbose-Logging-Lock*	319



8.65.1	Variable	319
8.66	Tootsville:*Weather-Kernel*	320
8.66.1	Variable	320
8.67	Tootsville:*Websocket-Server*	321
8.67.1	Variable	321
8.68	Tootsville:*Wind-Vector-Field*	322
8.68.1	Variable	322
8.69	Tootsville:*Ws-Chars-Broadcast*	323
8.69.1	Variable	323
8.70	Tootsville:*Ws-Chars-Received*	324
8.70.1	Variable	324
8.71	Tootsville:*Ws-Chars-Unicast*	325
8.71.1	Variable	325
8.72	Tootsville:*Ws-Client-For-Toot*	326
8.72.1	Variable	326
8.73	Tootsville:*Ws-Client-For-User*	327
8.73.1	Variable	327
8.74	Tootsville:*Ws-Connections*	328
8.74.1	Variable	328
8.75	Tootsville:*Ws-Sign-Ins*	329
8.75.1	Variable	329
8.76	Tootsville:*Ws-Surprise-Disconnects*	330
8.76.1	Variable	330
8.77	Tootsville:*Ws-Traffic-Commands*	331
8.77.1	Variable	331
8.78	Tootsville:*Ws-Traffic-From*	332
8.78.1	Variable	332
8.79	Tootsville:*Ws-Traffic-Other*	333
8.79.1	Variable	333
8.80	Tootsville::+Alexa-Timestamp-Tolerance+	334
8.80.1	Variable	334
8.81	Tootsville::+Amazon-Cert-Chain-Url-Matching+	335
8.81.1	Variable	335
8.82	Tootsville::+Backtrace-Regex+	336
8.82.1	Variable	336
8.83	Tootsville::+Builder-Toot-Hard-Hat-Template+	337
8.83.1	Variable	337
8.84	Tootsville::+Color24-Names+	338
8.84.1	Variable	338
8.85	Tootsville::+Color24-Values+	339
8.85.1	Variable	339
8.86	Tootsville::+Credits+	340
8.86.1	Variable	340
8.87	Tootsville::+Doc-Packages+	341
8.87.1	Variable	341
8.88	Tootsville::+Facing-Angles+	342
8.88.1	Variable	342
8.89	Tootsville::+Gravatar-Base-Uri+	343

8.89.1	Variable	343
8.90	Tootsville::+Habitat-Colors+	344
8.90.1	Variable	344
8.91	Tootsville::+Initial-T-Shirt-Colors+	345
8.91.1	Variable	345
8.92	Tootsville::+Moon-Day+	346
8.92.1	Variable	346
8.93	Tootsville::+Moon-Year+	347
8.93.1	Variable	347
8.94	Tootsville::+Other-Moon-Day+	348
8.94.1	Variable	348
8.95	Tootsville::+Other-Moon-Year+	349
8.95.1	Variable	349
8.96	Tootsville::+Pink-Moon-Day+	350
8.96.1	Variable	350
8.97	Tootsville::+Pink-Moon-Year+	351
8.97.1	Variable	351
8.98	Tootsville::+Pre-Login-Max-Commands+	352
8.98.1	Variable	352
8.99	Tootsville::+Pre-Login-Max-Time+	353
8.99.1	Variable	353
8.100	Tootsville::+Supported-Languages+	354
8.100.1	Variable	354
8.101	Tootsville::+Toot-Base-Color-Names+	355
8.101.1	Variable	355
8.102	Tootsville::+Toot-Basic-Pattern-Names+	356
8.102.1	Variable	356
8.103	Tootsville::+Toot-Extended-Pattern-Names+	357
8.103.1	Variable	357
8.104	Tootsville::+Toot-Pad-Color-Names+	358
8.104.1	Variable	358
8.105	Tootsville::+Toot-Pattern-Color-Names+	359
8.105.1	Variable	359
8.106	Tootsville::+Unix-Time-In-Universal+	360
8.106.1	Variable	360
8.107	Tootsville::+Unix-Zero-In-Universal-Time+	361
8.107.1	Variable	361
8.108	Tootsville::+Ws-Idle-Seconds+	362
8.108.1	Variable	362
8.109	Tootsville::2-Days-Ago	363
8.109.1	Function	363
8.109.2	File	363
8.110	Tootsville::3-Days-Ago	364
8.110.1	Function	364
8.110.2	File	364
8.111	Tootsville::@-Message	365
8.111.1	Function	365
8.111.2	File	365

8.112	Tootsville::Accept-Type-Equal	366
8.112.1	Function	366
8.112.2	File	366
8.113	Tootsville::Acceptor-Status-Message	367
8.113.1	Function	367
8.114	Tootsville::Accepts-Content-Type-P	368
8.114.1	Function	368
8.114.2	File	368
8.115	Tootsville::Active-Player	369
8.115.1	Function	369
8.115.2	File	369
8.116	Tootsville::Add-Charset	370
8.116.1	Function	370
8.116.2	File	370
8.117	Tootsville::Add-Contact	371
8.117.1	Function	371
8.117.2	File	371
8.118	Tootsville::Add-Or-Replace-Endpoint	372
8.118.1	Function	372
8.119	Tootsville::Admin-Message	373
8.119.1	Function	373
8.119.2	File	373
8.120	Tootsville::After-Slash	374
8.120.1	Function	374
8.120.2	File	374
8.121	Tootsville::All-Connected	375
8.121.1	Function	375
8.121.2	File	375
8.122	Tootsville::All-Credits	376
8.122.1	Function	376
8.122.2	File	376
8.123	Tootsville::All-Links-To-Same-Person-P	377
8.123.1	Function	377
8.123.2	File	377
8.124	Tootsville::All-Symbols-Alphabetically	378
8.124.1	Function	378
8.124.2	File	378
8.125	Tootsville::Allowed-Base-Colors-Under-Pattern	379
8.125.1	Function	379
8.125.2	File	379
8.126	Tootsville::Allowed-Pattern-Colors-On-Base	380
8.126.1	Function	380
8.126.2	File	380
8.127	Tootsville::Altitude	381
8.127.1	Function	381
8.127.2	File	381
8.128	Tootsville::Answered-Child-Requests-By-Toot	382
8.128.1	Function	382

8.128.2	File	382
8.129	Tootsville::Apply-Config	383
8.129.1	Function	383
8.129.2	File	383
8.130	Tootsville::Apply-Extension-To-Template	384
8.130.1	Function	384
8.130.2	File	384
8.131	Tootsville::Arrange-Columns+Values-For-Find	385
8.131.1	Function	385
8.131.2	File	385
8.132	Tootsville::Assert-My-Character	386
8.132.1	Function	386
8.132.2	File	386
8.133	Tootsville::Associate-Credentials	387
8.133.1	Function	387
8.133.2	File	387
8.134	Tootsville::Atom-Or-Comma-List	388
8.134.1	Function	388
8.134.2	File	388
8.135	Tootsville::Avatar	389
8.135.1	Class	389
8.135.2	Slots	389
8.136	Tootsville::Avatar-Avatar-Scale-X	390
8.136.1	Function	390
8.136.2	File	390
8.136.3	SetF Function	390
8.136.4	File	390
8.137	Tootsville::Avatar-Avatar-Scale-Y	391
8.137.1	Function	391
8.137.2	File	391
8.137.3	SetF Function	391
8.137.4	File	391
8.138	Tootsville::Avatar-Avatar-Scale-Z	392
8.138.1	Function	392
8.138.2	File	392
8.138.3	SetF Function	392
8.138.4	File	392
8.139	Tootsville::Avatar-Has-Slot-P	393
8.139.1	Function	393
8.139.2	File	393
8.140	Tootsville::Avatar-Id	394
8.140.1	Function	394
8.140.2	File	394
8.140.3	SetF Function	394
8.140.4	File	394
8.141	Tootsville::Avatar-Moniker	395
8.141.1	Function	395
8.141.2	File	395

8.141.3	SetF Function .....	395
8.141.4	File .....	395
8.142	Tootsville::Avatar-P .....	396
8.142.1	Function .....	396
8.142.2	File .....	396
8.143	Tootsville::Avatar-Slot .....	397
8.143.1	Class .....	397
8.143.2	Slots .....	397
8.144	Tootsville::Avatar-Slot-Avatar .....	398
8.144.1	Function .....	398
8.144.2	File .....	398
8.144.3	SetF Function .....	398
8.144.4	File .....	398
8.145	Tootsville::Avatar-Slot-Id .....	399
8.145.1	Function .....	399
8.145.2	File .....	399
8.145.3	SetF Function .....	399
8.145.4	File .....	399
8.146	Tootsville::Avatar-Slot-P .....	400
8.146.1	Function .....	400
8.146.2	File .....	400
8.147	Tootsville::Avatar-Slot-Slot .....	401
8.147.1	Function .....	401
8.147.2	File .....	401
8.147.3	SetF Function .....	401
8.147.4	File .....	401
8.148	Tootsville::Avatar-Slot-Valence .....	402
8.148.1	Function .....	402
8.148.2	File .....	402
8.148.3	SetF Function .....	402
8.148.4	File .....	402
8.149	Tootsville::Average .....	403
8.149.1	Function .....	403
8.149.2	File .....	403
8.150	Tootsville::Ayt-Idle-Users .....	404
8.150.1	Function .....	404
8.150.2	File .....	404
8.151	Tootsville::Background-Gc .....	405
8.151.1	Function .....	405
8.151.2	File .....	405
8.152	Tootsville::Bad-Request .....	406
8.152.1	Class .....	406
8.152.2	Slots .....	406
8.153	Tootsville::Bad-Request-Thing .....	407
8.153.1	Function .....	407
8.153.2	SetF Function .....	407
8.154	Tootsville::Banhammer-Ip-Address .....	408
8.154.1	Function .....	408

8.154.2	File	408
8.155	Tootsville::Banner	409
8.155.1	Function	409
8.155.2	File	409
8.156	Tootsville::Banner/ Error-Output	410
8.156.1	Function	410
8.156.2	File	410
8.157	Tootsville::Banner/ Log	411
8.157.1	Function	411
8.157.2	File	411
8.158	Tootsville::Banner/ Query-IO	412
8.158.1	Function	412
8.158.2	File	412
8.159	Tootsville::Banner/ Standard-Output	413
8.159.1	Function	413
8.159.2	File	413
8.160	Tootsville::Banner/ Trace-Output	414
8.160.1	Function	414
8.160.2	File	414
8.161	Tootsville::Base64-From-Uri-Form	415
8.161.1	Function	415
8.161.2	File	415
8.162	Tootsville::Base64-To-Uuid	416
8.162.1	Function	416
8.162.2	File	416
8.163	Tootsville::Basic-8-Personality	417
8.163.1	Class	417
8.163.2	Slots	417
8.164	Tootsville::Before-Save-Normalize	418
8.164.1	Function	418
8.165	Tootsville::Bool-Sort	419
8.165.1	Function	419
8.165.2	File	419
8.166	Tootsville::Broadcast	420
8.166.1	Function	420
8.166.2	File	420
8.167	Tootsville::Build-Simple-Column-Query	421
8.167.1	Function	421
8.167.2	File	421
8.168	Tootsville::Build-Simple-Query	422
8.168.1	Function	422
8.168.2	File	422
8.169	Tootsville::Builder-Toot-P	423
8.169.1	Function	423
8.169.2	File	423
8.170	Tootsville::Burgeon-Quiesced-State	424
8.170.1	Function	424
8.170.2	File	424

8.171	Tootsville::Byte-Vector-To-Integer	425
8.171.1	Function	425
8.171.2	File	425
8.172	Tootsville::Bytes-Json	426
8.172.1	Function	426
8.172.2	File	426
8.173	Tootsville::Call-Infinity-From-Rest	427
8.173.1	Function	427
8.173.2	File	427
8.174	Tootsville::Call-Infinity-From-Stream	428
8.174.1	Function	428
8.174.2	File	428
8.175	Tootsville::Cassandra-Add-To-Blacklist	429
8.175.1	Function	429
8.175.2	File	429
8.176	Tootsville::Cassandra-Add-To-Redlist	430
8.176.1	Function	430
8.176.2	File	430
8.177	Tootsville::Cassandra-Boot	431
8.177.1	Function	431
8.177.2	File	431
8.178	Tootsville::Cassandra-Filter	432
8.178.1	Function	432
8.178.2	File	432
8.179	Tootsville::Cassandra-Obnoxious-Filter	433
8.179.1	Function	433
8.179.2	File	433
8.180	Tootsville::Cassandra-Remove-From-Blacklist	434
8.180.1	Function	434
8.180.2	File	434
8.181	Tootsville::Cassandra-Remove-From-Redlist	435
8.181.1	Function	435
8.181.2	File	435
8.182	Tootsville::Chaos-Personality	436
8.182.1	Class	436
8.182.2	Slots	436
8.183	Tootsville::Character-Music	437
8.183.1	Class	437
8.183.2	Slots	437
8.184	Tootsville::Character-Music-Music	438
8.184.1	Function	438
8.184.2	File	438
8.184.3	SetF Function	438
8.184.4	File	438
8.185	Tootsville::Character-Music-P	439
8.185.1	Function	439
8.185.2	File	439
8.186	Tootsville::Character-Music-Toot	440

8.186.1	Function	440
8.186.2	File	440
8.186.3	SetF Function	440
8.186.4	File	440
8.187	Tootsville::Chdir	441
8.187.1	Function	441
8.187.2	File	441
8.188	Tootsville::Check-Alexa	442
8.188.1	Function	442
8.188.2	File	442
8.189	Tootsville::Check-Alexa-Signature	443
8.189.1	Function	443
8.189.2	File	443
8.190	Tootsville::Check-Alexa-Signature-Cert-Chain-Url	444
8.190.1	Function	444
8.190.2	File	444
8.191	Tootsville::Check-Alexa-Timestamp-Tolerance	445
8.191.1	Function	445
8.191.2	File	445
8.192	Tootsville::Check-Arg-Type	446
8.192.1	Macro	446
8.192.2	File	446
8.193	Tootsville::Check-Cert-Chain-Valid	447
8.193.1	Function	447
8.193.2	File	447
8.194	Tootsville::Check-Cert-Dates-Valid	448
8.194.1	Function	448
8.194.2	File	448
8.195	Tootsville::Check-Firebase-Id-Token	449
8.195.1	Function	449
8.195.2	File	449
8.196	Tootsville::Check-Pattern-On-Base-Color	450
8.196.1	Function	450
8.196.2	File	450
8.197	Tootsville::Check-Toot-Name	451
8.197.1	Function	451
8.197.2	File	451
8.198	Tootsville::Check-X.509-San	452
8.198.1	Function	452
8.198.2	File	452
8.199	Tootsville::Child-Code	453
8.199.1	Type	453
8.200	Tootsville::Child-Request	454
8.200.1	Class	454
8.200.2	Slots	454
8.201	Tootsville::Child-Request-Allowed-At	455
8.201.1	Function	455
8.201.2	File	455



8.201.3	SetF Function	455
8.201.4	File	455
8.202	Tootsville::Child-Request-Allowed-For	456
8.202.1	Function	456
8.202.2	File	456
8.202.3	SetF Function	456
8.202.4	File	456
8.203	Tootsville::Child-Request-Allowed-Until	457
8.203.1	Function	457
8.203.2	File	457
8.204	Tootsville::Child-Request-Denied-At	458
8.204.1	Function	458
8.204.2	File	458
8.204.3	SetF Function	458
8.204.4	File	458
8.205	Tootsville::Child-Request-P	459
8.205.1	Function	459
8.205.2	File	459
8.206	Tootsville::Child-Request-Placed-At	460
8.206.1	Function	460
8.206.2	File	460
8.206.3	SetF Function	460
8.206.4	File	460
8.207	Tootsville::Child-Request-Response	461
8.207.1	Function	461
8.207.2	File	461
8.207.3	SetF Function	461
8.207.4	File	461
8.208	Tootsville::Child-Request-Toot	462
8.208.1	Function	462
8.208.2	File	462
8.208.3	SetF Function	462
8.208.4	File	462
8.209	Tootsville::Child-Request-Uuid	463
8.209.1	Function	463
8.209.2	File	463
8.209.3	SetF Function	463
8.209.4	File	463
8.210	Tootsville::Clean-Ice-Credentials	464
8.210.1	Function	464
8.210.2	File	464
8.211	Tootsville::Clean-Symbols	465
8.211.1	Function	465
8.211.2	File	465
8.212	Tootsville::Clear-All-Endpoints	466
8.212.1	Function	466
8.212.2	File	466
8.213	Tootsville::Clouds	467

8.213.1	Function .....	467
8.213.2	File .....	467
8.214	Tootsville::Cluster .....	468
8.214.1	Function .....	468
8.214.2	File .....	468
8.215	Tootsville::Cluster-Name .....	469
8.215.1	Function .....	469
8.215.2	File .....	469
8.216	Tootsville::Cluster-Net-Name .....	470
8.216.1	Function .....	470
8.216.2	File .....	470
8.217	Tootsville::Cluster-Wide-Lock-Busy-Error .....	471
8.217.1	Class .....	471
8.217.2	Slots .....	471
8.218	Tootsville::Cluster-Wide-Lock-Busy-Warning .....	472
8.218.1	Class .....	472
8.218.2	Slots .....	472
8.219	Tootsville::Cluster-Wide-Lock-Condition .....	473
8.219.1	Class .....	473
8.219.2	Slots .....	473
8.220	Tootsville::Cluster-Wide-Lock-Error .....	474
8.220.1	Class .....	474
8.220.2	Slots .....	474
8.221	Tootsville::Cluster-Wide-Lock-Not-Locked .....	475
8.221.1	Class .....	475
8.221.2	Slots .....	475
8.222	Tootsville::Cluster-Wide-Lock-Not-Ours .....	476
8.222.1	Class .....	476
8.222.2	Slots .....	476
8.223	Tootsville::Color24 .....	477
8.223.1	Class .....	477
8.223.2	Slots .....	477
8.224	Tootsville::Color24-Blue .....	478
8.224.1	Function .....	478
8.224.2	File .....	478
8.224.3	SetF Function .....	478
8.224.4	File .....	478
8.225	Tootsville::Color24-Green .....	479
8.225.1	Function .....	479
8.225.2	File .....	479
8.225.3	SetF Function .....	479
8.225.4	File .....	479
8.226	Tootsville::Color24-Hsv .....	480
8.226.1	Function .....	480
8.226.2	File .....	480
8.227	Tootsville::Color24-Hue .....	481
8.227.1	Function .....	481
8.227.2	File .....	481

8.228	Tootsville::Color24-Name	482
8.228.1	Function	482
8.228.2	File	482
8.229	Tootsville::Color24-P	483
8.229.1	Function	483
8.229.2	File	483
8.230	Tootsville::Color24-Red	484
8.230.1	Function	484
8.230.2	File	484
8.230.3	SetF Function	484
8.230.4	File	484
8.231	Tootsville::Color24-Rgb	485
8.231.1	Function	485
8.231.2	File	485
8.232	Tootsville::Color24-Saturation	486
8.232.1	Function	486
8.232.2	File	486
8.233	Tootsville::Color24-To-Integer	487
8.233.1	Function	487
8.233.2	File	487
8.234	Tootsville::Color24-Value	488
8.234.1	Function	488
8.234.2	File	488
8.235	Tootsville::Color24/ =	489
8.235.1	Function	489
8.235.2	File	489
8.236	Tootsville::Color24=	490
8.236.1	Function	490
8.236.2	File	490
8.237	Tootsville::Column-Load-Mapping	491
8.237.1	Function	491
8.237.2	File	491
8.238	Tootsville::Column-Load-Value	492
8.238.1	Function	492
8.238.2	File	492
8.239	Tootsville::Column-Normalizer	493
8.239.1	Function	493
8.239.2	File	493
8.240	Tootsville::Column-Save-Mapping	494
8.240.1	Function	494
8.240.2	File	494
8.241	Tootsville::Column-Save-Value	495
8.241.1	Function	495
8.241.2	File	495
8.242	Tootsville::Compute-Fountain-Peanuts-For-Score	496
8.242.1	Function	496
8.242.2	File	496
8.243	Tootsville::Compute-Fountain-Random-Fairy-Dust	497

8.243.1	Function	497
8.243.2	File	497
8.244	Tootsville::Compute-Next-Keys-Update	498
8.244.1	Function	498
8.244.2	File	498
8.245	Tootsville::Concat	499
8.245.1	Function	499
8.245.2	File	499
8.246	Tootsville::Condition-Name	500
8.246.1	Function	500
8.246.2	File	500
8.247	Tootsville::Condition-Slots	501
8.247.1	Function	501
8.247.2	File	501
8.248	Tootsville::Config	502
8.248.1	Function	502
8.248.2	File	502
8.249	Tootsville::Connect-Cache	503
8.249.1	Function	503
8.249.2	File	503
8.250	Tootsville::Connect-Databases	504
8.250.1	Function	504
8.250.2	File	504
8.251	Tootsville::Connect-Maria	505
8.251.1	Function	505
8.251.2	File	505
8.252	Tootsville::Connect-Time	506
8.252.1	Function	506
8.253	Tootsville::Connected-Toot-Names	507
8.253.1	Function	507
8.253.2	File	507
8.254	Tootsville::Connected-Toots	508
8.254.1	Function	508
8.254.2	File	508
8.255	Tootsville::Consider-Child-Kick	509
8.255.1	Function	509
8.255.2	File	509
8.256	Tootsville::Constituentp	510
8.256.1	Function	510
8.256.2	File	510
8.257	Tootsville::Contact	511
8.257.1	Class	511
8.257.2	Slots	511
8.258	Tootsville::Contact-Added	512
8.258.1	Function	512
8.258.2	File	512
8.258.3	SetF Function	512
8.258.4	File	512

8.259	Tootsville::Contact-Contact .....	513
8.259.1	Function .....	513
8.259.2	File .....	513
8.259.3	SetF Function .....	513
8.259.4	File .....	513
8.260	Tootsville::Contact-Last-Used .....	514
8.260.1	Function .....	514
8.260.2	File .....	514
8.260.3	SetF Function .....	514
8.260.4	File .....	514
8.261	Tootsville::Contact-Owner .....	515
8.261.1	Function .....	515
8.261.2	File .....	515
8.261.3	SetF Function .....	515
8.261.4	File .....	515
8.262	Tootsville::Contact-P .....	516
8.262.1	Function .....	516
8.262.2	File .....	516
8.263	Tootsville::Contact-Starredp .....	517
8.263.1	Function .....	517
8.263.2	File .....	517
8.263.3	SetF Function .....	517
8.263.4	File .....	517
8.264	Tootsville::Contact-Uuid .....	518
8.264.1	Function .....	518
8.264.2	File .....	518
8.264.3	SetF Function .....	518
8.264.4	File .....	518
8.265	Tootsville::Contents-To-Bytes .....	519
8.265.1	Function .....	519
8.265.2	File .....	519
8.266	Tootsville::Copy-Avatar .....	520
8.266.1	Function .....	520
8.266.2	File .....	520
8.267	Tootsville::Copy-Avatar-Slot .....	521
8.267.1	Function .....	521
8.267.2	File .....	521
8.268	Tootsville::Copy-Character-Music .....	522
8.268.1	Function .....	522
8.268.2	File .....	522
8.269	Tootsville::Copy-Child-Request .....	523
8.269.1	Function .....	523
8.269.2	File .....	523
8.270	Tootsville::Copy-Color24 .....	524
8.270.1	Function .....	524
8.270.2	File .....	524
8.271	Tootsville::Copy-Contact .....	525
8.271.1	Function .....	525

8.271.2	File	525
8.272	Tootsville::Copy-Credential	526
8.272.1	Function	526
8.272.2	File	526
8.273	Tootsville::Copy-Game-Point	527
8.273.1	Function	527
8.273.2	File	527
8.274	Tootsville::Copy-Gossip-Initiation	528
8.274.1	Function	528
8.274.2	File	528
8.275	Tootsville::Copy-Inventory-Item	529
8.275.1	Function	529
8.275.2	File	529
8.276	Tootsville::Copy-Item	530
8.276.1	Function	530
8.276.2	File	530
8.277	Tootsville::Copy-Item-Template	531
8.277.1	Function	531
8.277.2	File	531
8.278	Tootsville::Copy-Locale-Music	532
8.278.1	Function	532
8.278.2	File	532
8.279	Tootsville::Copy-Login	533
8.279.1	Function	533
8.279.2	File	533
8.280	Tootsville::Copy-Lot	534
8.280.1	Function	534
8.280.2	File	534
8.281	Tootsville::Copy-Metronome-Task	535
8.281.1	Function	535
8.281.2	File	535
8.282	Tootsville::Copy-Mist	536
8.282.1	Function	536
8.282.2	File	536
8.283	Tootsville::Copy-Music	537
8.283.1	Function	537
8.283.2	File	537
8.284	Tootsville::Copy-Parent-Child	538
8.284.1	Function	538
8.284.2	File	538
8.285	Tootsville::Copy-Pattern	539
8.285.1	Function	539
8.285.2	File	539
8.286	Tootsville::Copy-Person	540
8.286.1	Function	540
8.286.2	File	540
8.287	Tootsville::Copy-Person-Link	541
8.287.1	Function	541

8.287.2	File	541
8.288	Tootsville::Copy-Place	542
8.288.1	Function	542
8.288.2	File	542
8.289	Tootsville::Copy-Quaestor-Event	543
8.289.1	Function	543
8.289.2	File	543
8.290	Tootsville::Copy-Sms	544
8.290.1	Function	544
8.290.2	File	544
8.291	Tootsville::Copy-Store-Item	545
8.291.1	Function	545
8.291.2	File	545
8.292	Tootsville::Copy-Tcp-Client	546
8.292.1	Function	546
8.292.2	File	546
8.293	Tootsville::Copy-Terrain-Edge-Horz	547
8.293.1	Function	547
8.293.2	File	547
8.294	Tootsville::Copy-Terrain-Edge-Vert	548
8.294.1	Function	548
8.294.2	File	548
8.295	Tootsville::Copy-Terrain-Height	549
8.295.1	Function	549
8.295.2	File	549
8.296	Tootsville::Copy-Toot	550
8.296.1	Function	550
8.296.2	File	550
8.297	Tootsville::Copy-Toot-Quiesced	551
8.297.1	Function	551
8.297.2	File	551
8.298	Tootsville::Copy-Wear-Slot	552
8.298.1	Function	552
8.298.2	File	552
8.299	Tootsville::Copy-Wind-Vector	553
8.299.1	Function	553
8.299.2	File	553
8.300	Tootsville::Copy-World	554
8.300.1	Function	554
8.300.2	File	554
8.301	Tootsville::Copy-Wtl-Course	555
8.301.1	Function	555
8.301.2	File	555
8.302	Tootsville::Create-Item	556
8.302.1	Function	556
8.302.2	File	556
8.303	Tootsville::Credential	557
8.303.1	Class	557

8.303.2	Slots .....	557
8.304	Tootsville::Credential-Auth-Token .....	558
8.304.1	Function .....	558
8.304.2	File .....	558
8.304.3	SetF Function .....	558
8.304.4	File .....	558
8.305	Tootsville::Credential-Id-Token .....	559
8.305.1	Function .....	559
8.305.2	File .....	559
8.305.3	SetF Function .....	559
8.305.4	File .....	559
8.306	Tootsville::Credential-Json-Info .....	560
8.306.1	Function .....	560
8.306.2	File .....	560
8.306.3	SetF Function .....	560
8.306.4	File .....	560
8.307	Tootsville::Credential-P .....	561
8.307.1	Function .....	561
8.307.2	File .....	561
8.308	Tootsville::Credential-Person .....	562
8.308.1	Function .....	562
8.308.2	File .....	562
8.308.3	SetF Function .....	562
8.308.4	File .....	562
8.309	Tootsville::Credential-Provider .....	563
8.309.1	Function .....	563
8.309.2	File .....	563
8.309.3	SetF Function .....	563
8.309.4	File .....	563
8.310	Tootsville::Credential-Refresh-Token .....	564
8.310.1	Function .....	564
8.310.2	File .....	564
8.310.3	SetF Function .....	564
8.310.4	File .....	564
8.311	Tootsville::Credential-Uid .....	565
8.311.1	Function .....	565
8.311.2	File .....	565
8.311.3	SetF Function .....	565
8.311.4	File .....	565
8.312	Tootsville::Credential-Uuid .....	566
8.312.1	Function .....	566
8.312.2	File .....	566
8.312.3	SetF Function .....	566
8.312.4	File .....	566
8.313	Tootsville::Cupid-Personality .....	567
8.313.1	Class .....	567
8.313.2	Slots .....	567
8.314	Tootsville::Current-Position .....	568



8.314.1	Function .....	568
8.315	Tootsville::Current-Temp .....	569
8.315.1	Function .....	569
8.315.2	File .....	569
8.316	Tootsville::Database-For .....	570
8.316.1	Function .....	570
8.316.2	File .....	570
8.317	Tootsville::Db-Config .....	571
8.317.1	Function .....	571
8.317.2	File .....	571
8.318	Tootsville::Db-Select .....	572
8.318.1	Function .....	572
8.318.2	File .....	572
8.319	Tootsville::Db-Select-All .....	573
8.319.1	Function .....	573
8.319.2	File .....	573
8.320	Tootsville::Db-Select-Records-Simply .....	574
8.320.1	Function .....	574
8.320.2	File .....	574
8.321	Tootsville::Db-Select-Single-Column .....	575
8.321.1	Function .....	575
8.321.2	File .....	575
8.322	Tootsville::Db-Select-Single-Record .....	576
8.322.1	Function .....	576
8.322.2	File .....	576
8.323	Tootsville::Db-Table-For .....	577
8.323.1	Function .....	577
8.323.2	File .....	577
8.324	Tootsville::Debugger .....	578
8.324.1	Function .....	578
8.324.2	File .....	578
8.325	Tootsville::Decode-Message .....	579
8.325.1	Function .....	579
8.325.2	File .....	579
8.326	Tootsville::Decorate-Endpoint-Template-Html .....	580
8.326.1	Function .....	580
8.326.2	File .....	580
8.327	Tootsville::Decorate-Method-Html .....	581
8.327.1	Function .....	581
8.327.2	File .....	581
8.328	Tootsville::Default-Config-File .....	582
8.328.1	Function .....	582
8.328.2	File .....	582
8.329	Tootsville::Defendpoint .....	583
8.329.1	Macro .....	583
8.329.2	File .....	583
8.330	Tootsville::Defendpoint/ Make-Docstring .....	584
8.330.1	Function .....	584

8.330.2	File	584
8.331	Tootsville::Defendpoint/ Make-Endpoint-Function	585
8.331.1	Function	585
8.331.2	File	585
8.332	Tootsville::Define-Alexa-Endpoint	586
8.332.1	Macro	586
8.332.2	File	586
8.333	Tootsville::Define-Character	587
8.333.1	Macro	587
8.333.2	File	587
8.334	Tootsville::Define-Maintenance-Task	588
8.334.1	Macro	588
8.334.2	File	588
8.335	Tootsville::Define-Operator-Command	589
8.335.1	Macro	589
8.335.2	File	589
8.336	Tootsville::Define-Personality	590
8.336.1	Macro	590
8.336.2	File	590
8.337	Tootsville::Define-Reply	591
8.337.1	Macro	591
8.337.2	File	591
8.338	Tootsville::Definfinity	592
8.338.1	Macro	592
8.338.2	History of Infinity Mode	592
8.338.3	Wire protocols	592
8.338.3.1	RESTful POSTs	592
8.338.4	Datagram constructions	593
8.338.5	logOK datagrams	593
8.338.6	Command datagrams	593
8.338.7	Gatekeeper datagrams	594
8.338.8	File	594
8.339	Tootsville::Defpost	595
8.339.1	Macro	595
8.339.2	File	595
8.340	Tootsville::Defrecord	596
8.340.1	Macro	596
8.340.2	File	596
8.341	Tootsville::Defrecord/ Before-Save-Normalize	597
8.341.1	Function	597
8.341.2	File	597
8.342	Tootsville::Defrecord/ Column-To-Json-Pair	598
8.342.1	Function	598
8.342.2	File	598
8.343	Tootsville::Defrecord/ Destroy-Record	599
8.343.1	Function	599
8.343.2	File	599
8.344	Tootsville::Defrecord/ Find-Record	600

8.344.1	Function .....	600
8.344.2	File .....	600
8.345	Tootsville::Defrecord/ Find-Record/ Pull .....	601
8.345.1	Function .....	601
8.345.2	File .....	601
8.346	Tootsville::Defrecord/ Find-Records .....	602
8.346.1	Function .....	602
8.346.2	File .....	602
8.347	Tootsville::Defrecord/ Find-Records-By-Sql .....	603
8.347.1	Function .....	603
8.347.2	File .....	603
8.348	Tootsville::Defrecord/ Find-Records/ Pull .....	604
8.348.1	Function .....	604
8.348.2	File .....	604
8.349	Tootsville::Defrecord/ Find-Reference .....	605
8.349.1	Function .....	605
8.349.2	File .....	605
8.350	Tootsville::Defrecord/ Find-Reference-Columns .....	606
8.350.1	Function .....	606
8.350.2	File .....	606
8.351	Tootsville::Defrecord/ Id-Column-For .....	607
8.351.1	Function .....	607
8.351.2	File .....	607
8.352	Tootsville::Defrecord/ Invalidate-Cache .....	608
8.352.1	Function .....	608
8.352.2	File .....	608
8.353	Tootsville::Defrecord/ Load-Record .....	609
8.353.1	Function .....	609
8.353.2	File .....	609
8.354	Tootsville::Defrecord/ Make-Record .....	610
8.354.1	Function .....	610
8.354.2	File .....	610
8.355	Tootsville::Defrecord/ Record= .....	611
8.355.1	Function .....	611
8.355.2	File .....	611
8.356	Tootsville::Defrecord/ Save-Record .....	612
8.356.1	Function .....	612
8.356.2	File .....	612
8.357	Tootsville::Defrecord/ Save-Record-With-Id-Column .....	613
8.357.1	Function .....	613
8.357.2	File .....	613
8.358	Tootsville::Defrecord/ To-Json .....	614
8.358.1	Function .....	614
8.358.2	File .....	614
8.359	Tootsville::Delete-Contact .....	615
8.359.1	Function .....	615
8.359.2	File .....	615
8.360	Tootsville::Demand-Quiesce-Toot .....	616

8.360.1	Function	616
8.360.2	File	616
8.361	Tootsville::Describe-System	617
8.361.1	Function	617
8.361.2	Example Output	617
8.361.3	File	617
8.362	Tootsville::Describe-World	618
8.362.1	Function	618
8.362.2	File	618
8.363	Tootsville::Destroy-All-Idle-Workers	619
8.363.1	Function	619
8.363.2	File	619
8.364	Tootsville::Destroy-All-Listeners	620
8.364.1	Function	620
8.364.2	File	620
8.365	Tootsville::Destroy-All-Web-Tasks	621
8.365.1	Function	621
8.365.2	File	621
8.366	Tootsville::Destroy-Endpoint	622
8.366.1	Function	622
8.366.2	File	622
8.367	Tootsville::Destroy-Record	623
8.367.1	Function	623
8.367.2	File	623
8.368	Tootsville::Destroy-Toot	624
8.368.1	Function	624
8.368.2	File	624
8.369	Tootsville::Detailed-Time	625
8.369.1	Function	625
8.369.2	File	625
8.370	Tootsville::Devel	626
8.370.1	Variable	626
8.371	Tootsville::Disable-Sbcl-Ldb	627
8.371.1	Function	627
8.371.2	File	627
8.372	Tootsville::Disconnect-No-Login	628
8.372.1	Function	628
8.372.2	File	628
8.373	Tootsville::Dispatch-Endpoint	629
8.373.1	Function	629
8.373.2	File	629
8.374	Tootsville::Distance	630
8.374.1	Function	630
8.374.2	File	630
8.375	Tootsville::Divisible-By-200-P	631
8.375.1	Function	631
8.375.2	File	631
8.376	Tootsville::Dns-Name	632

8.376.1	Type	632
8.377	Tootsville::Do-After	633
8.377.1	Macro	633
8.377.2	File	633
8.378	Tootsville::Do-Db-Records-Simply	634
8.378.1	Macro	634
8.378.2	File	634
8.379	Tootsville::Do-Metronome	635
8.379.1	Macro	635
8.379.2	File	635
8.380	Tootsville::Do-Records	636
8.380.1	Macro	636
8.380.2	File	636
8.381	Tootsville::Docstring->Html	637
8.381.1	Function	637
8.381.2	File	637
8.382	Tootsville::Docstring->Markdown	638
8.382.1	Function	638
8.382.2	File	638
8.383	Tootsville::Doff-Any-Conflicting-Item	639
8.383.1	Function	639
8.383.2	File	639
8.384	Tootsville::Doff-Item	640
8.384.1	Function	640
8.384.2	File	640
8.385	Tootsville::Doff-Item-In-Slot	641
8.385.1	Function	641
8.385.2	File	641
8.386	Tootsville::Don-Item	642
8.386.1	Function	642
8.386.2	File	642
8.387	Tootsville::Doodle-Personality	643
8.387.1	Class	643
8.387.2	Slots	643
8.388	Tootsville::Dottie-Personality	644
8.388.1	Class	644
8.388.2	Slots	644
8.389	Tootsville::Double-@	645
8.389.1	Function	645
8.389.2	File	645
8.390	Tootsville::Drop-Item	646
8.390.1	Function	646
8.390.2	File	646
8.391	Tootsville::Dump-Credits	647
8.391.1	Function	647
8.391.2	File	647
8.392	Tootsville::Dump-Global-Heightmap	648
8.392.1	Function	648

8.392.2	File	648
8.393	Tootsville::Email-Lhs	649
8.393.1	Function	649
8.393.2	File	649
8.394	Tootsville::Enable-Sbcl-Ldb	650
8.394.1	Function	650
8.394.2	File	650
8.395	Tootsville::Enable-Ssl-P	651
8.395.1	Function	651
8.395.2	File	651
8.396	Tootsville::Encode-Endpoint-Reply	652
8.396.1	Function	652
8.396.2	File	652
8.397	Tootsville::Endpoint	653
8.397.1	Class	653
8.397.2	Slots	653
8.398	Tootsville::Endpoint->Html	654
8.398.1	Function	654
8.398.2	File	654
8.399	Tootsville::Endpoint->Openapi	655
8.399.1	Function	655
8.399.2	File	655
8.400	Tootsville::Endpoint->Plist	656
8.400.1	Function	656
8.400.2	File	656
8.401	Tootsville::Endpoint-Close	657
8.401.1	Function	657
8.401.2	File	657
8.402	Tootsville::Endpoint-Close-Key	658
8.402.1	Function	658
8.402.2	File	658
8.403	Tootsville::Endpoint-Content-Type	659
8.403.1	Function	659
8.404	Tootsville::Endpoint-Delete-/ Users/ Me/ Toots/ Toot-Name⇨Json	660
8.404.1	Function	660
8.404.2	Status: 202 Toot deletion in progress	660
8.404.3	Status: 204 Toot deleted	660
8.404.4	Status: 401 Authorization Required	660
8.404.5	Status: 403 Authorization Failed	660
8.404.6	Status: 404 Not Found	660
8.404.7	Status: 405 Not Allowed	660
8.404.8	Web Service Endpoint	660
8.404.9	File	660
8.405	Tootsville::Endpoint-Function	661
8.405.1	Function	661
8.406	Tootsville::Endpoint-Get-/ Favicon/ Ico⇨Vnd.Microsoft.Icon	662
8.406.1	Function	662

8.406.2	Web Service Endpoint	662
8.406.3	File	662
8.407	Tootsville::Endpoint-Get-/ Favicon→Gif	663
8.407.1	Function	663
8.407.2	Web Service Endpoint	663
8.407.3	File	663
8.408	Tootsville::Endpoint-Get-/ Favicon→Png	664
8.408.1	Function	664
8.408.2	Web Service Endpoint	664
8.408.3	File	664
8.409	Tootsville::Endpoint-Get-/ Gossip/ Answers/ Uuid→Sdp	665
8.409.1	Function	665
8.409.2	204 No Content	665
8.409.3	200 OK	665
8.409.4	Web Service Endpoint	665
8.409.5	File	665
8.410	Tootsville::Endpoint-Get-/ Gossip/ Ice-Servers→Json	666
8.410.1	Function	666
8.410.2	Web Service Endpoint	666
8.410.3	File	666
8.411	Tootsville::Endpoint-Get-/ Gossip/ Offers→Json	667
8.411.1	Function	667
8.411.2	Web Service Endpoint	667
8.411.3	File	667
8.412	Tootsville::Endpoint-Get-/ Maintenance/ →Txt	668
8.412.1	Function	668
8.412.2	Web Service Endpoint	668
8.412.3	File	668
8.413	Tootsville::Endpoint-Get-/ Meta-Game/ Headers→Json	669
8.413.1	Function	669
8.413.2	Web Service Endpoint	669
8.413.3	File	669
8.414	Tootsville::Endpoint-Get-/ Meta-Game/ Ping→Txt	670
8.414.1	Function	670
8.414.2	200: OK	670
8.414.3	Web Service Endpoint	670
8.414.4	File	670
8.415	Tootsville::Endpoint-Get-/ Meta-Game/ Services/ Old→Json	671
8.415.1	Function	671
8.415.2	Web Service Endpoint	671
8.415.3	File	671
8.416	Tootsville::Endpoint-Get-/ Meta-Game/ Services→Html	672
8.416.1	Function	672
8.416.2	Web Service Endpoint	672
8.416.3	File	672
8.417	Tootsville::Endpoint-Get-/ Meta-Game/ Services→Json	673
8.417.1	Function	673
8.417.2	Status: 200 OK	673

8.417.3	Web Service Endpoint	673
8.417.4	File	673
8.418	Tootsville::Endpoint-Get-/ Toots/ Toot-Name→Json	674
8.418.1	Function	674
8.418.2	200 OK	674
8.418.3	404 Not Found	674
8.418.4	Web Service Endpoint	674
8.418.5	File	674
8.419	Tootsville::Endpoint-Get-/ Toots/ Toot-Name→Txt	675
8.419.1	Function	675
8.419.2	Web Service Endpoint	675
8.419.3	File	675
8.420	Tootsville::Endpoint-Get-/ Users/ Me/ Toots/ Toot-Name→Json	676
8.420.1	Function	676
8.420.2	Status: 200 OK	676
8.420.3	Status: 401 Authorization Required	676
8.420.4	Status: 403 Authorization Failed	676
8.420.5	Status: 404 Not Found	676
8.420.6	Web Service Endpoint	676
8.420.7	File	676
8.421	Tootsville::Endpoint-Get-/ Users/ Me/ Toots/ Toot-Name→Txt	677
8.421.1	Function	677
8.421.2	Web Service Endpoint	677
8.421.3	File	677
8.422	Tootsville::Endpoint-Get-/ Users/ Me/ Toots→Json	678
8.422.1	Function	678
8.422.2	200 OK	678
8.422.3	Web Service Endpoint	678
8.422.4	File	678
8.423	Tootsville::Endpoint-Get-/ Users/ Me/ Toots→Txt	679
8.423.1	Function	679
8.423.2	Web Service Endpoint	679
8.423.3	File	679
8.424	Tootsville::Endpoint-Get-/ Users/ Me→Json	680
8.424.1	Function	680
8.424.2	Status: 200 OK	680
8.424.3	Status: 401 Authorization Required	680
8.424.4	Status: 403 Authorization Failed	680
8.424.5	Web Service Endpoint	680
8.424.6	File	680
8.425	Tootsville::Endpoint-Get-/ Users/ Me→Txt	681
8.425.1	Function	681
8.425.2	Web Service Endpoint	681
8.425.3	File	681
8.426	Tootsville::Endpoint-Get-/ Version/ About/ Detail/ Param→Json	682



8.426.1	Function .....	682
8.426.2	Web Service Endpoint.....	682
8.426.3	File .....	682
8.427	Tootsville::Endpoint-Get-/ Version/ About/ Detail/ Param→Txt .....	683
8.427.1	Function .....	683
8.427.2	Web Service Endpoint.....	684
8.427.3	File .....	684
8.428	Tootsville::Endpoint-Get-/ Version/ About→Json.....	685
8.428.1	Function .....	685
8.428.2	Web Service Endpoint.....	685
8.428.3	File .....	685
8.429	Tootsville::Endpoint-Get-/ Version/ About→Txt .....	686
8.429.1	Function .....	686
8.429.2	Web Service Endpoint.....	686
8.429.3	File .....	686
8.430	Tootsville::Endpoint-Get-/ World/ Clock/ Calendar/ Now/ Fragment→Html .....	687
8.430.1	Function .....	687
8.430.2	Web Service Endpoint.....	687
8.430.3	File .....	687
8.431	Tootsville::Endpoint-Get-/ World/ Clock/ Calendar/ Year/ Year/ Fragment→Html .....	688
8.431.1	Function .....	688
8.431.2	Web Service Endpoint.....	688
8.431.3	File .....	688
8.432	Tootsville::Endpoint-Get-/ World/ Clock/ Calendar/ Year/ Year/ Month/ Month/ Fragment→Html .....	689
8.432.1	Function .....	689
8.432.2	Web Service Endpoint.....	689
8.432.3	File .....	689
8.433	Tootsville::Endpoint-Get-/ World/ Clock/ Calendar/ Year/ Year/ Month/ Month→Html .....	690
8.433.1	Function .....	690
8.433.2	Web Service Endpoint.....	690
8.433.3	File .....	690
8.434	Tootsville::Endpoint-Get-/ World/ Clock/ Date/ Abbrev→Txt..	691
8.434.1	Function .....	691
8.434.2	Web Service Endpoint.....	691
8.434.3	File .....	691
8.435	Tootsville::Endpoint-Get-/ World/ Clock/ Date/ Long→Txt..	692
8.435.1	Function .....	692
8.435.2	Web Service Endpoint.....	692
8.435.3	File .....	692
8.436	Tootsville::Endpoint-Get-/ World/ Clock/ Date→Txt.....	693
8.436.1	Function .....	693
8.436.2	Web Service Endpoint.....	693
8.436.3	File .....	693

- 8.437 Tootsville::Endpoint-Get-/ World/  
 Clock/ Time/ Detailed→Txt ..... 694
  - 8.437.1 Function ..... 694
  - 8.437.2 Web Service Endpoint ..... 694
  - 8.437.3 File ..... 694
- 8.438 Tootsville::Endpoint-Get-/ World/ Clock/ Time→Json ..... 695
  - 8.438.1 Function ..... 695
  - 8.438.2 Web Service Endpoint ..... 695
  - 8.438.3 File ..... 695
- 8.439 Tootsville::Endpoint-Get-/ World/ Clock/ Time→Txt ..... 696
  - 8.439.1 Function ..... 696
  - 8.439.2 Web Service Endpoint ..... 696
  - 8.439.3 File ..... 696
- 8.440 Tootsville::Endpoint-Get-/ World/ Sky/  
 Tootanga/ Latitude/ Longitude→Json ..... 697
  - 8.440.1 Function ..... 697
  - 8.440.2 Web Service Endpoint ..... 697
  - 8.440.3 File ..... 697
- 8.441 Tootsville::Endpoint-Get-/ World/ Tootanga/  
 Latitude/ Longitude/ Altitude→Json ..... 698
  - 8.441.1 Function ..... 698
  - 8.441.2 Web Service Endpoint ..... 698
  - 8.441.3 File ..... 698
- 8.442 Tootsville::Endpoint-Get-/ World→Json ..... 699
  - 8.442.1 Function ..... 699
  - 8.442.2 Web Service Endpoint ..... 699
  - 8.442.3 File ..... 699
- 8.443 Tootsville::Endpoint-Get-/ →Html ..... 700
  - 8.443.1 Function ..... 700
  - 8.443.2 Web Service Endpoint ..... 700
  - 8.443.3 File ..... 700
- 8.444 Tootsville::Endpoint-Hash ..... 701
  - 8.444.1 Function ..... 701
  - 8.444.2 File ..... 701
- 8.445 Tootsville::Endpoint-Kinda-Key ..... 702
  - 8.445.1 Function ..... 702
  - 8.445.2 File ..... 702
- 8.446 Tootsville::Endpoint-Method ..... 703
  - 8.446.1 Function ..... 703
- 8.447 Tootsville::Endpoint-Patch-/ Users/ Me→Json ..... 704
  - 8.447.1 Function ..... 704
  - 8.447.2 Status: 200 OK ..... 704
  - 8.447.3 Status: 401 Authorization Required ..... 704
  - 8.447.4 Status: 403 Authorization Failed ..... 704
  - 8.447.5 Status: 405 Not Allowed ..... 704
  - 8.447.6 Web Service Endpoint ..... 704
  - 8.447.7 File ..... 704

8.448	Tootsville::Endpoint-Post-/ Gossip/ Alexa/ Chat/ Region/ Region→Json .....	705
8.448.1	Function .....	705
8.448.2	Web Service Endpoint .....	705
8.448.3	File .....	705
8.449	Tootsville::Endpoint-Post-/ Gossip/ Alexa/ Clock/ Region/ Region→Json .....	706
8.449.1	Function .....	706
8.449.2	Web Service Endpoint .....	706
8.449.3	File .....	706
8.450	Tootsville::Endpoint-Post-/ Gossip/ Alexa/ Info/ Region/ Region→Json .....	707
8.450.1	Function .....	707
8.450.2	Web Service Endpoint .....	707
8.450.3	File .....	707
8.451	Tootsville::Endpoint-Post-/ Gossip/ Answers/ Uuid→Sdp... ..	708
8.451.1	Function .....	708
8.451.2	202 Accepted .....	708
8.451.3	404 Not Found .....	708
8.451.4	Web Service Endpoint .....	708
8.451.5	File .....	708
8.452	Tootsville::Endpoint-Post-/ Gossip/ Offers→Sdp .....	709
8.452.1	Function .....	709
8.452.2	Web Service Endpoint .....	709
8.452.3	File .....	709
8.453	Tootsville::Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Call→Xml .....	710
8.453.1	Function .....	710
8.453.2	Web Service Endpoint .....	710
8.453.3	File .....	710
8.454	Tootsville::Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Fax→Xml .....	711
8.454.1	Function .....	711
8.454.2	Web Service Endpoint .....	711
8.454.3	File .....	711
8.455	Tootsville::Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Sms→Xml .....	712
8.455.1	Function .....	712
8.455.2	Web Service Endpoint .....	712
8.455.3	File .....	712
8.456	Tootsville::Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Verify→Xml .....	713
8.456.1	Function .....	713
8.456.2	Web Service Endpoint .....	713
8.456.3	File .....	713
8.457	Tootsville::Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Whatsapp→Xml .....	714
8.457.1	Function .....	714

8.457.2	Web Service Endpoint	714
8.457.3	File	714
8.458	Tootsville::Endpoint-Post-/ Login/ Child→Json	715
8.458.1	Function	715
8.458.2	Web Service Endpoint	715
8.458.3	File	715
8.459	Tootsville::Endpoint-Post-/ Maintenance/ Buildapp/ Status→Nil	716
8.459.1	Function	716
8.459.2	Web Service Endpoint	716
8.459.3	File	716
8.460	Tootsville::Endpoint-Post-/ Maintenance/ Buildapp→Nil	717
8.460.1	Function	717
8.460.2	Web Service Endpoint	717
8.460.3	File	717
8.461	Tootsville::Endpoint-Post-/ Maintenance/ Hot-Reload→Nil	718
8.461.1	Function	718
8.461.2	Web Service Endpoint	718
8.461.3	File	718
8.462	Tootsville::Endpoint-Post-/ Maintenance/ Quicklisp-Update→Nil	719
8.462.1	Function	719
8.462.2	Web Service Endpoint	719
8.462.3	File	719
8.463	Tootsville::Endpoint-Post-/ Maintenance/ Quit→Nil	720
8.463.1	Function	720
8.463.2	Web Service Endpoint	720
8.463.3	File	720
8.464	Tootsville::Endpoint-Post-/ Maintenance/ Reload-Jscl→Nil	721
8.464.1	Function	721
8.464.2	Web Service Endpoint	721
8.464.3	File	721
8.465	Tootsville::Endpoint-Post-/ Toots→Json	722
8.465.1	Function	722
8.465.2	Web Service Endpoint	722
8.465.3	File	722
8.466	Tootsville::Endpoint-Post-/ Users/ Me/ Play-With/ Toot-Name→Json	723
8.466.1	Function	723
8.466.2	Status: 200 OK	723
8.466.3	Status: 401 Authorization Required	723
8.466.4	Status: 403 Authorization Failed	723
8.466.5	Status: 404 Not Found	723
8.466.6	Status: 405 Not Allowed	723
8.466.7	Web Service Endpoint	723
8.466.8	File	723
8.467	Tootsville::Endpoint-Post-/ World/ Infinity/ Add-Furniture→Json	724

8.467.1	Function	724
8.467.2	Infinity Mode command	724
8.467.3	Web Service Endpoint	724
8.467.4	File	724
8.468	Tootsville::Endpoint-Post-/ World/ Infinity/ Add-Journal-Entry→Json	725
8.468.1	Function	725
8.468.2	Infinity Mode command	725
8.468.3	Web Service Endpoint	725
8.468.4	File	725
8.469	Tootsville::Endpoint-Post-/ World/ Infinity/ Add-To-List→Json	726
8.469.1	Function	726
8.469.2	Infinity Mode command	726
8.469.3	Web Service Endpoint	726
8.469.4	File	726
8.470	Tootsville::Endpoint-Post-/ World/ Infinity/ Click→Json	727
8.470.1	Function	727
8.470.2	Infinity Mode command	727
8.470.3	Web Service Endpoint	727
8.470.4	File	727
8.471	Tootsville::Endpoint-Post-/ World/ Infinity/ Consider-Child-Approval→Json	728
8.471.1	Function	728
8.471.2	Infinity Mode command	728
8.471.3	Web Service Endpoint	728
8.471.4	File	728
8.472	Tootsville::Endpoint-Post-/ World/ Infinity/ Create-User-House→Json	729
8.472.1	Function	729
8.472.2	Infinity Mode command	729
8.472.3	Web Service Endpoint	729
8.472.4	File	729
8.473	Tootsville::Endpoint-Post-/ World/ Infinity/ Delete-Mail-Message→Json	730
8.473.1	Function	730
8.473.2	Infinity Mode command	730
8.473.3	Web Service Endpoint	730
8.473.4	File	730
8.474	Tootsville::Endpoint-Post-/ World/ Infinity/ Doff→Json	731
8.474.1	Function	731
8.474.2	Infinity Mode command	731
8.474.3	Web Service Endpoint	731
8.474.4	File	731
8.475	Tootsville::Endpoint-Post-/ World/ Infinity/ Doffi→Json	732
8.475.1	Function	732
8.475.2	Infinity Mode command	732
8.475.3	Web Service Endpoint	732

8.475.4	File .....	732
8.476	Tootsville::Endpoint-Post-/ World/ Infinity/ Don→Json ....	733
8.476.1	Function .....	733
8.476.2	Infinity Mode command .....	733
8.476.3	Web Service Endpoint.....	733
8.476.4	File .....	733
8.477	Tootsville::Endpoint-Post-/ World/ Infinity/ Echo→Json ...	734
8.477.1	Function .....	734
8.477.2	Infinity Mode command .....	734
8.477.3	Web Service Endpoint.....	734
8.477.4	File .....	734
8.478	Tootsville::Endpoint-Post-/ World/ Infinity/ End-Event→Json..	735
8.478.1	Function .....	735
8.478.2	Infinity Mode command .....	735
8.478.3	Web Service Endpoint.....	735
8.478.4	File .....	735
8.479	Tootsville::Endpoint-Post-/ World/ Infinity/ Enumerate-Wear-Slots→Json .....	736
8.479.1	Function .....	736
8.479.2	Infinity Mode command .....	736
8.479.3	Web Service Endpoint.....	736
8.479.4	File .....	736
8.480	Tootsville::Endpoint-Post-/ World/ Infinity/ Finger→Json..	737
8.480.1	Function .....	737
8.480.2	Infinity Mode command .....	737
8.480.3	Web Service Endpoint.....	737
8.480.4	File .....	737
8.481	Tootsville::Endpoint-Post-/ World/ Infinity/ Game-Action→Json .....	738
8.481.1	Function .....	738
8.481.2	Infinity Mode command .....	738
8.481.3	Web Service Endpoint.....	738
8.481.4	File .....	738
8.482	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Avatars→Json .....	739
8.482.1	Function .....	739
8.482.2	Infinity Mode command .....	739
8.482.3	Web Service Endpoint.....	739
8.482.4	File .....	739
8.483	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Color-Palettes→Json.....	740
8.483.1	Function .....	740
8.483.2	Infinity Mode command .....	740
8.483.3	Web Service Endpoint.....	740
8.483.4	File .....	740
8.484	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Inventory-By-Type→Json .....	741
8.484.1	Function .....	741

8.484.2	Infinity Mode command .....	741
8.484.3	Web Service Endpoint .....	741
8.484.4	File .....	741
8.485	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Inventory→Json .....	742
8.485.1	Function .....	742
8.485.2	Infinity Mode command .....	742
8.485.3	Web Service Endpoint .....	742
8.485.4	File .....	742
8.486	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Mail-In-Box→Json .....	743
8.486.1	Function .....	743
8.486.2	Infinity Mode command .....	743
8.486.3	Web Service Endpoint .....	743
8.486.4	File .....	743
8.487	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Online-Users→Json .....	744
8.487.1	Function .....	744
8.487.2	Infinity Mode command .....	744
8.487.3	Web Service Endpoint .....	744
8.487.4	File .....	744
8.488	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Passport→Json .....	745
8.488.1	Function .....	745
8.488.2	Infinity Mode command .....	745
8.488.3	Web Service Endpoint .....	745
8.488.4	File .....	745
8.489	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Room-List→Json .....	746
8.489.1	Function .....	746
8.489.2	Infinity Mode command .....	746
8.489.3	Web Service Endpoint .....	746
8.489.4	File .....	746
8.490	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Room-Vars→Json .....	747
8.490.1	Function .....	747
8.490.2	Infinity Mode command .....	747
8.490.3	Web Service Endpoint .....	747
8.490.4	File .....	747
8.491	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Server-Time→Json .....	748
8.491.1	Function .....	748
8.491.2	Infinity Mode command .....	748
8.491.3	Web Service Endpoint .....	748
8.491.4	File .....	748
8.492	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Session-Apple→Json .....	749
8.492.1	Function .....	749

8.492.2	Infinity Mode command .....	749
8.492.3	Web Service Endpoint .....	749
8.492.4	File .....	749
8.493	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Store-Item-Info→Json .....	750
8.493.1	Function .....	750
8.493.2	Infinity Mode command .....	750
8.493.3	Web Service Endpoint .....	750
8.493.4	File .....	750
8.494	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-User-Lists→Json .....	751
8.494.1	Function .....	751
8.494.2	Infinity Mode command .....	751
8.494.3	Web Service Endpoint .....	751
8.494.4	File .....	751
8.495	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Wallet→Json ..	752
8.495.1	Function .....	752
8.495.2	Infinity Mode command .....	752
8.495.3	Web Service Endpoint .....	752
8.495.4	File .....	752
8.496	Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Zone-List→Json .....	753
8.496.1	Function .....	753
8.496.2	Infinity Mode command .....	753
8.496.3	Web Service Endpoint .....	753
8.496.4	File .....	753
8.497	Tootsville::Endpoint-Post-/ World/ Infinity/ Give→Json ....	754
8.497.1	Function .....	754
8.497.2	Infinity Mode command .....	754
8.497.3	Web Service Endpoint .....	754
8.497.4	File .....	754
8.498	Tootsville::Endpoint-Post-/ World/ Infinity/ Go→Json ....	755
8.498.1	Function .....	755
8.498.2	Infinity Mode command .....	755
8.498.3	Web Service Endpoint .....	755
8.498.4	File .....	755
8.499	Tootsville::Endpoint-Post-/ World/ Infinity/ Init-User-Room→Json .....	756
8.499.1	Function .....	756
8.499.2	Infinity Mode command .....	756
8.499.3	Web Service Endpoint .....	756
8.499.4	File .....	756
8.500	Tootsville::Endpoint-Post-/ World/ Infinity/ Join→Json ....	757
8.500.1	Function .....	757
8.500.2	Infinity Mode command .....	757
8.500.3	Web Service Endpoint .....	757
8.500.4	File .....	757
8.501	Tootsville::Endpoint-Post-/ World/ Infinity/ Logout→Json ..	758



8.501.1	Function .....	758
8.501.2	Infinity Mode command .....	758
8.501.3	Web Service Endpoint .....	758
8.501.4	File .....	758
8.502	Tootsville::Endpoint-Post-/ World/ Infinity/ Mail-Customer-Service→Json .....	759
8.502.1	Function .....	759
8.502.2	Infinity Mode command .....	759
8.502.3	Web Service Endpoint .....	759
8.502.4	File .....	759
8.503	Tootsville::Endpoint-Post-/ World/ Infinity/ Peek-At-Inventory→Json .....	760
8.503.1	Function .....	760
8.503.2	Infinity Mode command .....	760
8.503.3	Web Service Endpoint .....	760
8.503.4	File .....	760
8.504	Tootsville::Endpoint-Post-/ World/ Infinity/ Ping→Json .....	761
8.504.1	Function .....	761
8.504.2	Infinity Mode command .....	761
8.504.3	Web Service Endpoint .....	761
8.504.4	File .....	761
8.505	Tootsville::Endpoint-Post-/ World/ Infinity/ Play-With→Json .....	762
8.505.1	Function .....	762
8.505.2	Infinity Mode command .....	762
8.505.3	Web Service Endpoint .....	762
8.505.4	File .....	762
8.506	Tootsville::Endpoint-Post-/ World/ Infinity/ Prompt-Reply→Json .....	763
8.506.1	Function .....	763
8.506.2	Infinity Mode command .....	763
8.506.3	Web Service Endpoint .....	763
8.506.4	File .....	763
8.507	Tootsville::Endpoint-Post-/ World/ Infinity/ Quiesce→Json .....	764
8.507.1	Function .....	764
8.507.2	Infinity Mode command .....	764
8.507.3	Web Service Endpoint .....	764
8.507.4	File .....	764
8.508	Tootsville::Endpoint-Post-/ World/ Infinity/ Read-Map→Json .....	765
8.508.1	Function .....	765
8.508.2	Infinity Mode command .....	765
8.508.3	Web Service Endpoint .....	765
8.508.4	File .....	765
8.509	Tootsville::Endpoint-Post-/ World/ Infinity/ Remove-From-List→Json .....	766
8.509.1	Function .....	766
8.509.2	Infinity Mode command .....	766
8.509.3	Web Service Endpoint .....	766
8.509.4	File .....	766

8.510	Tootsville::Endpoint-Post-/ World/ Infinity/ Report-Bug→Json .....	767
8.510.1	Function .....	767
8.510.2	Infinity Mode command .....	767
8.510.3	Web Service Endpoint .....	767
8.510.4	File .....	767
8.511	Tootsville::Endpoint-Post-/ World/ Infinity/ Report-User→Json .....	768
8.511.1	Function .....	768
8.511.2	Infinity Mode command .....	768
8.511.3	Web Service Endpoint .....	768
8.511.4	File .....	768
8.512	Tootsville::Endpoint-Post-/ World/ Infinity/ Request-Buddy→Json .....	769
8.512.1	Function .....	769
8.512.2	Infinity Mode command .....	769
8.512.3	Web Service Endpoint .....	769
8.512.4	File .....	769
8.513	Tootsville::Endpoint-Post-/ World/ Infinity/ Send-Mail-Message→Json .....	770
8.513.1	Function .....	770
8.513.2	Infinity Mode command .....	770
8.513.3	Web Service Endpoint .....	770
8.513.4	File .....	770
8.514	Tootsville::Endpoint-Post-/ World/ Infinity/ Send-Out-Of-Band-Message→Json .....	771
8.514.1	Function .....	771
8.514.2	Infinity Mode command .....	771
8.514.3	Web Service Endpoint .....	771
8.514.4	File .....	771
8.515	Tootsville::Endpoint-Post-/ World/ Infinity/ Server-Time→Json .....	772
8.515.1	Function .....	772
8.515.2	Infinity Mode command .....	772
8.515.3	Web Service Endpoint .....	772
8.515.4	File .....	772
8.516	Tootsville::Endpoint-Post-/ World/ Infinity/ Set-Avatar-Color→Json .....	773
8.516.1	Function .....	773
8.516.2	Infinity Mode command .....	773
8.516.3	Web Service Endpoint .....	773
8.516.4	File .....	773
8.517	Tootsville::Endpoint-Post-/ World/ Infinity/ Set-Furniture→Json .....	774
8.517.1	Function .....	774
8.517.2	Infinity Mode command .....	774
8.517.3	Web Service Endpoint .....	774
8.517.4	File .....	774

8.518	Tootsville::Endpoint-Post-/ World/ Infinity/ Set-Room-Var→Json .....	775
8.518.1	Function .....	775
8.518.2	Infinity Mode command .....	775
8.518.3	Web Service Endpoint .....	775
8.518.4	File .....	775
8.519	Tootsville::Endpoint-Post-/ World/ Infinity/ Set-User-Var→Json .....	776
8.519.1	Function .....	776
8.519.2	Infinity Mode command .....	776
8.519.3	Web Service Endpoint .....	776
8.519.4	File .....	776
8.520	Tootsville::Endpoint-Post-/ World/ Infinity/ Shoot→Json ..	777
8.520.1	Function .....	777
8.520.2	Infinity Mode command .....	777
8.520.3	Web Service Endpoint .....	777
8.520.4	File .....	777
8.521	Tootsville::Endpoint-Post-/ World/ Infinity/ Spawn-Zone→Json .....	778
8.521.1	Function .....	778
8.521.2	Infinity Mode command .....	778
8.521.3	Web Service Endpoint .....	778
8.521.4	File .....	778
8.522	Tootsville::Endpoint-Post-/ World/ Infinity/ Speak→Json ..	779
8.522.1	Function .....	779
8.522.2	Infinity Mode command .....	779
8.522.3	Web Service Endpoint .....	779
8.522.4	File .....	779
8.523	Tootsville::Endpoint-Post-/ World/ Infinity/ Stamp-Passport→Json .....	780
8.523.1	Function .....	780
8.523.2	Infinity Mode command .....	780
8.523.3	Web Service Endpoint .....	780
8.523.4	File .....	780
8.524	Tootsville::Endpoint-Post-/ World/ Infinity/ Start-Event→Json .....	781
8.524.1	Function .....	781
8.524.2	Infinity Mode command .....	781
8.524.3	Web Service Endpoint .....	781
8.524.4	File .....	781
8.525	Tootsville::Endpoint-Post-/ World/ Infinity/ Toot-List→Json ..	782
8.525.1	Function .....	782
8.525.2	Infinity Mode command .....	782
8.525.3	Web Service Endpoint .....	782
8.525.4	File .....	782
8.526	Tootsville::Endpoint-Post-/ World/ Infinity/ Use-Equipment→Json .....	783
8.526.1	Function .....	783

8.526.2	Infinity Mode command .....	783
8.526.3	Web Service Endpoint .....	783
8.526.4	File .....	783
8.527	Tootsville::Endpoint-Post-/ World/ Infinity/ Wardrobe→Json..	784
8.527.1	Function .....	784
8.527.2	Infinity Mode command .....	784
8.527.3	Web Service Endpoint .....	784
8.527.4	File .....	784
8.528	Tootsville::Endpoint-Post-/ World/ Infinity/ Wtl-4→Json...	785
8.528.1	Function .....	785
8.528.2	Infinity Mode command .....	785
8.528.3	Web Service Endpoint .....	785
8.528.4	File .....	785
8.529	Tootsville::Endpoint-Post-/ World/ Infinity/ Wtl→Json ....	786
8.529.1	Function .....	786
8.529.2	Infinity Mode command .....	786
8.529.3	Web Service Endpoint .....	786
8.529.4	File .....	786
8.530	Tootsville::Endpoint-Post-/ World/ Infinity→Json .....	787
8.530.1	Function .....	787
8.530.2	Web Service Endpoint .....	787
8.530.3	File .....	787
8.531	Tootsville::Endpoint-Put-/ Toots/ Toot-Name→Json .....	788
8.531.1	Function .....	788
8.531.2	Web Service Endpoint .....	788
8.531.3	File .....	788
8.532	Tootsville::Endpoint-Put-/ Users/ Me→Json .....	789
8.532.1	Function .....	789
8.532.2	Status: 201 Created .....	789
8.532.3	Status: 401 Authorization Required .....	789
8.532.4	Status: 403 Authorization Failed .....	789
8.532.5	Status: 405 Not Allowed .....	789
8.532.6	Status: 422 .....	789
8.532.7	Web Service Endpoint .....	789
8.532.8	File .....	789
8.533	Tootsville::Endpoint-Template .....	790
8.533.1	Function .....	790
8.534	Tootsville::Endpoint-Template-Arity .....	791
8.534.1	Function .....	791
8.535	Tootsville::Endpoint-Template-Match .....	792
8.535.1	Function .....	792
8.535.2	File .....	792
8.536	Tootsville::Endpoint-Template-String .....	793
8.536.1	Function .....	793
8.536.2	File .....	793
8.537	Tootsville::Endpoint-Vars->Openapi .....	794
8.537.1	Function .....	794
8.537.2	File .....	794

8.538	Tootsville::Endpoints-Equal	795
8.538.1	Function	795
8.538.2	File	795
8.539	Tootsville::Endpoints-Page-Footer	796
8.539.1	Function	796
8.539.2	File	796
8.540	Tootsville::Endpoints-Page-Header	797
8.540.1	Function	797
8.540.2	File	797
8.541	Tootsville::Endpoints-Prefixed	798
8.541.1	Function	798
8.541.2	File	798
8.542	Tootsville::Ensure-Integer	799
8.542.1	Function	799
8.542.2	File	799
8.543	Tootsville::Ensure-Inventory-Item	800
8.543.1	Function	800
8.544	Tootsville::Ensure-List-Of-People	801
8.544.1	Function	801
8.544.2	File	801
8.545	Tootsville::Ensure-Message-Is-Characters	802
8.545.1	Function	802
8.545.2	File	802
8.546	Tootsville::Ensure-Number	803
8.546.1	Function	803
8.546.2	File	803
8.547	Tootsville::Ensure-Record	804
8.547.1	Function	804
8.547.2	File	804
8.548	Tootsville::Ensure-Site-Name	805
8.548.1	Function	805
8.548.2	File	805
8.549	Tootsville::Ensure-Toot	806
8.549.1	Function	806
8.549.2	File	806
8.550	Tootsville::Ensure-User-For-Plist	807
8.550.1	Function	807
8.550.2	File	807
8.551	Tootsville::Ensure-Wear-Slot	808
8.551.1	Function	808
8.552	Tootsville::Ensure-Weather-Kernel	809
8.552.1	Function	809
8.552.2	File	809
8.553	Tootsville::Entry	810
8.553.1	Function	810
8.553.2	File	810
8.554	Tootsville::Enumerate-Endpoints	811
8.554.1	Function	811

8.554.2	File	811
8.555	Tootsville::Erase-All-Memcached-For	812
8.555.1	Function	812
8.555.2	File	812
8.556	Tootsville::Error-Log-File	813
8.556.1	Function	813
8.556.2	File	813
8.557	Tootsville::Every-Toot-Name	814
8.557.1	Function	814
8.557.2	File	814
8.558	Tootsville::Extension-For-Content-Type	815
8.558.1	Function	815
8.558.2	File	815
8.559	Tootsville::Extract-Certificate-Base64	816
8.559.1	Function	816
8.559.2	File	816
8.560	Tootsville::Extract-Plist-Path	817
8.560.1	Function	817
8.560.2	File	817
8.561	Tootsville::Extract-Public-Key-From-Cert	818
8.561.1	Function	818
8.561.2	File	818
8.562	Tootsville::Fetch-Ice-Credentials/ Xirsys	819
8.562.1	Function	819
8.562.2	File	819
8.563	Tootsville::Fetch-Json	820
8.563.1	Function	820
8.563.2	File	820
8.564	Tootsville::Fill-Blank-Contour	821
8.564.1	Function	821
8.564.2	File	821
8.565	Tootsville::Find-Acceptor	822
8.565.1	Function	822
8.565.2	File	822
8.566	Tootsville::Find-Best-Endpoint	823
8.566.1	Function	823
8.566.2	File	823
8.567	Tootsville::Find-Client-For-Socket	824
8.567.1	Function	824
8.567.2	File	824
8.568	Tootsville::Find-Exact-Endpoint	825
8.568.1	Function	825
8.568.2	File	825
8.569	Tootsville::Find-Infinity-Websocket-Resource	826
8.569.1	Function	826
8.569.2	File	826
8.570	Tootsville::Find-Kinda-Endpoint	827
8.570.1	Function	827

8.570.2	File	827
8.571	Tootsville::Find-Log-Dir	828
8.571.1	Function	828
8.571.2	File	828
8.572	Tootsville::Find-Person-By-Url	829
8.572.1	Function	829
8.572.2	File	829
8.573	Tootsville::Find-Player-Or-Die	830
8.573.1	Function	830
8.573.2	File	830
8.574	Tootsville::Find-Random-Point-If	831
8.574.1	Function	831
8.574.2	File	831
8.575	Tootsville::Find-Record	832
8.575.1	Function	832
8.575.2	File	832
8.576	Tootsville::Find-Records	833
8.576.1	Function	833
8.576.2	File	833
8.577	Tootsville::Find-Records-By-Sql	834
8.577.1	Function	834
8.577.2	File	834
8.578	Tootsville::Find-Reference	835
8.578.1	Function	835
8.578.2	File	835
8.579	Tootsville::Find-Results-In-Docstring	836
8.579.1	Function	836
8.579.2	File	836
8.580	Tootsville::Find-Robot	837
8.580.1	Function	837
8.580.2	File	837
8.581	Tootsville::Find-Thread	838
8.581.1	Function	838
8.581.2	File	838
8.582	Tootsville::Find-Toot-By-Name	839
8.582.1	Function	839
8.582.2	File	839
8.583	Tootsville::Find-Toot-Passport	840
8.583.1	Function	840
8.583.2	File	840
8.584	Tootsville::Find-User-For-Credentials	841
8.584.1	Function	841
8.584.2	File	841
8.585	Tootsville::Find-User-For-Headers	842
8.585.1	Function	842
8.585.2	File	842
8.586	Tootsville::Find-User-For-Json	843
8.586.1	Function	843

8.586.2	File .....	843
8.587	Tootsville::Find-Var-In-Docstring .....	844
8.587.1	Function .....	844
8.587.2	File .....	844
8.588	Tootsville::First-Line .....	845
8.588.1	Function .....	845
8.588.2	File .....	845
8.589	Tootsville::First-Paragraph .....	846
8.589.1	Function .....	846
8.589.2	File .....	846
8.590	Tootsville::Flatten-Plist-Tree .....	847
8.590.1	Function .....	847
8.590.2	File .....	847
8.591	Tootsville::Flora-Personality .....	848
8.591.1	Class .....	848
8.591.2	Slots .....	848
8.592	Tootsville::Force-Close-Hunchensocket .....	849
8.592.1	Function .....	849
8.592.2	File .....	849
8.593	Tootsville::Fountain-Duplicate-P .....	850
8.593.1	Function .....	850
8.593.2	File .....	850
8.594	Tootsville::Fountain-Reject-As-Already-Done .....	851
8.594.1	Function .....	851
8.594.2	File .....	851
8.595	Tootsville::From-Avatars .....	852
8.595.1	Function .....	852
8.595.2	File .....	852
8.596	Tootsville::Game-Action-Bowling-Reset-Pins .....	853
8.596.1	Function .....	853
8.596.2	Usage .....	853
8.596.3	Effects .....	853
8.596.4	File .....	853
8.597	Tootsville::Game-Action-Bowling-Strike-Pins .....	854
8.597.1	Function .....	854
8.597.2	Usage .....	854
8.597.3	Effects .....	854
8.597.4	File .....	854
8.598	Tootsville::Game-Action-Card-Game-Arrange .....	855
8.598.1	Function .....	855
8.598.2	Usage .....	855
8.598.3	Effects .....	855
8.598.4	File .....	855
8.599	Tootsville::Game-Action-Card-Game-Deal .....	856
8.599.1	Function .....	856
8.599.2	Usage .....	856
8.599.3	Effects .....	856
8.599.4	File .....	856



8.600	Tootsville::Game-Action-Card-Game-Draw	857
8.600.1	Function	857
8.600.2	Usage	857
8.600.3	Effects	857
8.600.4	File	857
8.601	Tootsville::Game-Action-Card-Game-Move	858
8.601.1	Function	858
8.601.2	Usage	858
8.601.3	Effects	858
8.601.4	File	858
8.602	Tootsville::Game-Action-Card-Game-Play	859
8.602.1	Function	859
8.602.2	Usage	859
8.602.3	Effects	859
8.602.4	File	859
8.603	Tootsville::Game-Action-Card-Game-Shuffle	860
8.603.1	Function	860
8.603.2	Usage	860
8.603.3	Effects	860
8.603.4	File	860
8.604	Tootsville::Game-Action-Card-Game-Take	861
8.604.1	Function	861
8.604.2	Usage	861
8.604.3	Effects	861
8.604.4	File	861
8.605	Tootsville::Game-Action-Get-Bowling-Scorecard	862
8.605.1	Function	862
8.605.2	Usage	862
8.605.3	Effects	862
8.605.4	File	862
8.606	Tootsville::Game-Action-Join-Bowling-Game	863
8.606.1	Function	863
8.606.2	Usage	863
8.606.3	Effects	863
8.606.4	File	863
8.607	Tootsville::Game-Action-Join-Card-Game	864
8.607.1	Function	864
8.607.2	Usage	864
8.607.3	Overview of Card Games	864
8.607.4	Joining a Card Game	864
8.607.5	Usage	865
8.607.6	Effects	865
8.607.7	File	865
8.608	Tootsville::Game-Action-Part-Bowling-Game	866
8.608.1	Function	866
8.608.2	Usage	866
8.608.3	Effects	866
8.608.4	File	866

8.609	Tootsville::Game-Action-Part-Card-Game	867
8.609.1	Function	867
8.609.2	Usage	867
8.609.3	Effects	867
8.609.4	File	867
8.610	Tootsville::Game-Action-Pause-Sports-Ball-Timer	868
8.610.1	Function	868
8.610.2	Usage	868
8.610.3	Effects	868
8.610.4	File	868
8.611	Tootsville::Game-Action-Sports-Ball-Goal	869
8.611.1	Function	869
8.611.2	Usage	869
8.611.3	Example	869
8.611.4	Effects	869
8.611.5	File	869
8.612	Tootsville::Game-Action-Start-Bowling	870
8.612.1	Function	870
8.612.2	Usage	870
8.612.3	Effects	870
8.612.4	Overview of Bowling	870
8.612.5	Bowling gameAction actions	870
8.612.6	Starting a Bowling Game	870
8.612.7	File	870
8.613	Tootsville::Game-Action-Start-Sports-Ball-Game	871
8.613.1	Function	871
8.613.2	Usage	871
8.613.3	Example	871
8.613.4	Effects	871
8.613.5	About SportsBall	871
8.613.6	File	871
8.614	Tootsville::Game-Action-Start-Sports-Ball-Timer	872
8.614.1	Function	872
8.614.2	Usage	872
8.614.3	Effects	872
8.614.4	File	872
8.615	Tootsville::Game-Action-Tag-You-Re-It	873
8.615.1	Function	873
8.615.2	Usage	873
8.615.3	Effects	873
8.615.4	File	873
8.616	Tootsville::Game-Point	874
8.616.1	Class	874
8.616.2	Slots	874
8.617	Tootsville::Game-Point-Altitude	875
8.617.1	Function	875
8.617.2	File	875
8.617.3	SetF Function	875

8.617.4	File	875
8.618	Tootsville::Game-Point-Latitude	876
8.618.1	Function	876
8.618.2	File	876
8.618.3	SetF Function	876
8.618.4	File	876
8.619	Tootsville::Game-Point-Longitude	877
8.619.1	Function	877
8.619.2	File	877
8.619.3	SetF Function	877
8.619.4	File	877
8.620	Tootsville::Game-Point-P	878
8.620.1	Function	878
8.620.2	File	878
8.621	Tootsville::Game-Point-World	879
8.621.1	Function	879
8.621.2	File	879
8.621.3	SetF Function	879
8.621.4	File	879
8.622	Tootsville::Game-Point-X	880
8.622.1	Function	880
8.622.2	File	880
8.622.3	SetF Function	880
8.622.4	File	880
8.623	Tootsville::Game-Point-Y	881
8.623.1	Function	881
8.623.2	File	881
8.623.3	SetF Function	881
8.623.4	File	881
8.624	Tootsville::Game-Point-Z	882
8.624.1	Function	882
8.624.2	File	882
8.624.3	SetF Function	882
8.624.4	File	882
8.625	Tootsville::Gather-All-Symbols	883
8.625.1	Function	883
8.625.2	File	883
8.626	Tootsville::Generate-Blank-Contour	884
8.626.1	Function	884
8.626.2	File	884
8.627	Tootsville::Generate-Buddy-List-Signature	885
8.627.1	Function	885
8.627.2	File	885
8.628	Tootsville::Generate-Skydome-Cloud-Layer	886
8.628.1	Function	886
8.628.2	File	886
8.629	Tootsville::Generate-Terrain-Blank-Edge-Horz	887
8.629.1	Function	887

8.629.2	File	887
8.630	Tootsville::Generate-Terrain-Blank-Edge-Vert	888
8.630.1	Function	888
8.630.2	File	888
8.631	Tootsville::Generate-Terrain-Contour	889
8.631.1	Function	889
8.631.2	File	889
8.632	Tootsville::Generate-Terrain-Features	890
8.632.1	Function	890
8.632.2	File	890
8.633	Tootsville::Get-9-Terrain-Tiles	891
8.633.1	Function	891
8.633.2	File	891
8.634	Tootsville::Get-Google-Account-Keys	892
8.634.1	Function	892
8.634.2	File	892
8.635	Tootsville::Get-Last-Insert-Id	893
8.635.1	Function	893
8.635.2	File	893
8.636	Tootsville::Get-Mariadb-Lock	894
8.636.1	Function	894
8.636.2	File	894
8.637	Tootsville::Get-Rollbar-Person	895
8.637.1	Function	895
8.637.2	File	895
8.638	Tootsville::Get-Unix-Time	896
8.638.1	Function	896
8.638.2	File	896
8.639	Tootsville::Gift-Item	897
8.639.1	Function	897
8.639.2	File	897
8.640	Tootsville::Global-Heightmap-Corner	898
8.640.1	Function	898
8.640.2	File	898
8.640.3	SetF Function	898
8.640.4	File	898
8.641	Tootsville::Gone	899
8.641.1	Class	899
8.641.2	Slots	899
8.642	Tootsville::Gossip-Initiation	900
8.642.1	Class	900
8.642.2	Slots	900
8.643	Tootsville::Gossip-Initiation-Answer	901
8.643.1	Function	901
8.643.2	File	901
8.643.3	SetF Function	901
8.643.4	File	901
8.644	Tootsville::Gossip-Initiation-P	902

8.644.1	Function .....	902
8.644.2	File .....	902
8.645	Tootsville::Gossip-Initiation-Uuid .....	903
8.645.1	Function .....	903
8.645.2	File .....	903
8.645.3	SetF Function .....	903
8.645.4	File .....	903
8.646	Tootsville::Gracefully-Report-Error.Html .....	904
8.646.1	Function .....	904
8.646.2	File .....	904
8.647	Tootsville::Gracefully-Report-Error.Json .....	905
8.647.1	Function .....	905
8.647.2	File .....	905
8.648	Tootsville::Gracefully-Report-Http-Client-Error .....	906
8.648.1	Function .....	906
8.648.2	File .....	906
8.649	Tootsville::Grant-Item .....	907
8.649.1	Function .....	907
8.649.2	File .....	907
8.650	Tootsville::Gravatar-Hash .....	908
8.650.1	Function .....	908
8.650.2	File .....	908
8.651	Tootsville::Gravatar-Image-Url .....	909
8.651.1	Function .....	909
8.651.2	File .....	909
8.652	Tootsville::Greeting/ Daemon/ Error-Output .....	910
8.652.1	Function .....	910
8.652.2	File .....	910
8.653	Tootsville::Greeting/ Daemon/ Log-Output .....	911
8.653.1	Function .....	911
8.653.2	File .....	911
8.654	Tootsville::Greeting/ Daemon/ Standard-Output .....	912
8.654.1	Function .....	912
8.654.2	File .....	912
8.655	Tootsville::Greeting/ Daemon/ Trace-Output .....	913
8.655.1	Function .....	913
8.655.2	File .....	913
8.656	Tootsville::Group-Plists .....	914
8.656.1	Function .....	914
8.656.2	File .....	914
8.657	Tootsville::Habitat-Elevation-Roughness .....	915
8.657.1	Function .....	915
8.657.2	File .....	915
8.658	Tootsville::Habitat<-Pixel .....	916
8.658.1	Function .....	916
8.658.2	File .....	916
8.659	Tootsville::Handle-Normal-Request .....	917
8.659.1	Function .....	917

8.659.2	File	917
8.660	Tootsville::Handle-Options-Request	918
8.660.1	Function	918
8.660.2	File	918
8.661	Tootsville::Harmony-Personality	919
8.661.1	Class	919
8.661.2	Slots	919
8.662	Tootsville::Header-Time	920
8.662.1	Function	920
8.662.2	File	920
8.663	Tootsville::Holiday-Special-Personality	921
8.663.1	Class	921
8.663.2	Slots	921
8.664	Tootsville::Host-Name-Char-P	922
8.664.1	Function	922
8.664.2	File	922
8.665	Tootsville::Host-Name-Like-P	923
8.665.1	Function	923
8.665.2	File	923
8.666	Tootsville::How-Slow-Is-Slow	924
8.666.1	Function	924
8.667	Tootsville::Http-Client-Error	925
8.667.1	Class	925
8.667.2	Slots	925
8.668	Tootsville::Http-Idempotent-Request-Method	926
8.668.1	Type	926
8.669	Tootsville::Http-Is-Success-P	927
8.669.1	Function	927
8.669.2	File	927
8.670	Tootsville::Http-Request-Method	928
8.670.1	Type	928
8.671	Tootsville::Http-Safe-Request-Method	929
8.671.1	Type	929
8.672	Tootsville::Http-Status-Code	930
8.672.1	Function	930
8.673	Tootsville::Ice-Credentials	931
8.673.1	Function	931
8.673.2	File	931
8.674	Tootsville::Ice-Url-To-Urns	932
8.674.1	Function	932
8.674.2	File	932
8.675	Tootsville::Id-Column-For	933
8.675.1	Function	933
8.675.2	File	933
8.676	Tootsville::Ignore-Duplicates	934
8.676.1	Macro	934
8.676.2	File	934
8.677	Tootsville::Ignore-Not-Found	935

8.677.1	Macro	935
8.677.2	File	935
8.678	Tootsville::Infinity-Add-Furniture	936
8.678.1	Function	936
8.678.2	File	936
8.679	Tootsville::Infinity-Add-Journal-Entry	937
8.679.1	Function	937
8.679.2	Usage	937
8.679.3	Example	937
8.679.4	Romance 1.2 documentation	937
8.679.5	Formerly Proprietary Extension	937
8.679.6	File	937
8.680	Tootsville::Infinity-Add-To-List	938
8.680.1	Function	938
8.680.2	Usage	938
8.680.3	410 Gone	938
8.680.4	Changes from 1.1 to 1.2	938
8.680.5	File	938
8.681	Tootsville::Infinity-Click	939
8.681.1	Function	939
8.681.2	Usage	939
8.681.3	Modifiers characters	939
8.681.4	Flash details	940
8.681.5	Changes from 1.2 to 2.0	940
8.681.6	202 Accepted	940
8.681.7	204 No Content	940
8.681.8	File	940
8.682	Tootsville::Infinity-Consider-Child-Approval	941
8.682.1	Function	941
8.682.2	Usage	941
8.682.3	Error conditions	941
8.682.4	File	941
8.683	Tootsville::Infinity-Create-User-House	942
8.683.1	Function	942
8.683.2	Usage	942
8.683.3	201 Created	942
8.683.4	409 Conflict	942
8.683.5	404 Not Found	942
8.683.6	Changes from 1.2 to 2.0	942
8.683.7	Changes from 1.1 to 1.2	942
8.683.8	File	942
8.684	Tootsville::Infinity-Delete-Mail-Message	943
8.684.1	Function	943
8.684.2	Usage	943
8.684.3	Example	943
8.684.4	Changes from 1.2 to 2.0	943
8.684.5	Formerly Proprietary Extension	943
8.684.6	File	943

8.685	Tootsville::Infinity-Doff	944
8.685.1	Function	944
8.685.2	Usage	944
8.685.3	Example	944
8.685.4	400 Bad Request	944
8.685.5	200 OK	944
8.685.6	Formerly Proprietary Extension	944
8.685.7	File	944
8.686	Tootsville::Infinity-Dofff	945
8.686.1	Function	945
8.686.2	Usage	945
8.686.3	Limitations	945
8.686.4	Status 200 OK	945
8.686.5	File	945
8.687	Tootsville::Infinity-Don	946
8.687.1	Function	946
8.687.2	Usage	946
8.687.3	200 OK	946
8.687.4	400 Bad Request	946
8.687.5	404 Not Found	946
8.687.6	403 Forbidden	946
8.687.7	409 Conflict	946
8.687.8	Changes from 1.2 to 2.0	946
8.687.9	Changes from 1.0 to 1.1	947
8.687.10	File	947
8.688	Tootsville::Infinity-Echo	948
8.688.1	Function	948
8.688.2	Usage	948
8.688.2.1	Parameters	948
8.688.3	Example	948
8.688.4	200 OK	948
8.688.5	Limitations	948
8.688.6	Changes from 1.2 to 2.0	948
8.688.7	Known bugs	948
8.688.8	File	948
8.689	Tootsville::Infinity-End-Event	949
8.689.1	Function	949
8.689.2	Calling	949
8.689.3	Success Response to Canceled Event	949
8.689.4	Success Response to Completed Event	950
8.689.5	Error Responses	950
8.689.6	Changes from 1.2 to 2.0	951
8.689.7	File	951
8.690	Tootsville::Infinity-Enumerate-Wear-Slots	952
8.690.1	Function	952
8.690.2	Usage	952
8.690.3	200 OK	952
8.690.4	File	952



8.691	Tootsville::Infinity-Finger .....	953
8.691.1	Function .....	953
8.691.2	Usage .....	953
8.691.3	Reply format .....	953
8.691.4	File .....	953
8.692	Tootsville::Infinity-Game-Action .....	954
8.692.1	Function .....	954
8.692.2	Usage .....	954
8.692.3	Example .....	954
8.692.4	Overview of In-World Minigames .....	954
8.692.5	General Structure .....	954
8.692.6	Response format .....	955
8.692.7	Status 400 Error .....	955
8.692.8	File .....	955
8.693	Tootsville::Infinity-Get-Apple .....	956
8.693.1	Function .....	956
8.693.2	Theory .....	956
8.693.3	Apple-based authentication .....	956
8.693.4	New in 1.1 .....	957
8.693.5	Changes from 1.1 to 1.2 .....	958
8.693.6	Changes from 1.2 to 2.0 .....	958
8.693.7	File .....	958
8.694	Tootsville::Infinity-Get-Avatars .....	959
8.694.1	Function .....	959
8.694.2	Usage .....	959
8.694.3	Example .....	959
8.694.4	Status 200 OK .....	959
8.694.5	File .....	959
8.695	Tootsville::Infinity-Get-Color-Palettes .....	960
8.695.1	Function .....	960
8.695.2	Usage .....	960
8.695.3	Status 410 Gone .....	960
8.695.4	Changes from 1.1 to 1.2 .....	960
8.695.5	Revival? .....	960
8.695.6	File .....	960
8.696	Tootsville::Infinity-Get-Inventory .....	961
8.696.1	Function .....	961
8.696.2	Usage .....	961
8.696.3	Status 200 OK .....	961
8.696.4	File .....	961
8.697	Tootsville::Infinity-Get-Inventory-By-Type .....	962
8.697.1	Function .....	962
8.697.2	Usage .....	962
8.697.3	Changes from 1.2 to 2.0 .....	963
8.697.4	Status 200 OK .....	963
8.697.5	File .....	963
8.698	Tootsville::Infinity-Get-Mail-In-Box .....	964
8.698.1	Function .....	964

8.698.2	Usage	964
8.698.3	Examples	964
8.698.4	200 OK	964
8.698.5	416 Request Range Not Satisfiable	964
8.698.6	Changes from 1.2 to 2.0	964
8.698.7	Formerly Proprietary Extension	965
8.698.8	File	965
8.699	Tootsville::Infinity-Get-Online-Users	966
8.699.1	Function	966
8.699.2	Usage	966
8.699.3	Example	966
8.699.4	Status 200 OK	966
8.699.5	Status 403 Permission Denied	966
8.699.6	File	966
8.700	Tootsville::Infinity-Get-Passport	967
8.700.1	Function	967
8.700.2	Usage	967
8.700.3	200 OK	967
8.700.4	Changes from 1.2 to 2.0	967
8.700.5	Formerly Proprietary Extension	967
8.700.6	File	967
8.701	Tootsville::Infinity-Get-Room-List	968
8.701.1	Function	968
8.701.2	Usage	968
8.701.3	Status 200 OK	968
8.701.4	Changes from 1.2 to 2.0	968
8.701.5	File	968
8.702	Tootsville::Infinity-Get-Room-Vars	969
8.702.1	Function	969
8.702.2	Usage	969
8.702.3	Historical Usage (Romance I)	969
8.702.4	Room Environment	969
8.702.5	Sky Variables	969
8.702.6	Weather	969
8.702.7	Room Objects	969
8.702.7.1	Changes from 1.2 to 2.0	970
8.702.7.2	Changes from 1.1 to 1.2	971
8.702.8	Places	971
8.702.8.1	Changes from 1.2 to 2.0	972
8.702.8.2	Changes from 1.0 to 1.1	972
8.702.9	More good stuff	972
8.702.10	See Also	972
8.702.11	File	972
8.703	Tootsville::Infinity-Get-Server-Time	973
8.703.1	Function	973
8.703.2	File	973
8.704	Tootsville::Infinity-Get-Session-Apple	974
8.704.1	Function	974

8.704.2	410 Gone .....	974
8.704.3	New in 1.1 .....	974
8.704.4	File .....	974
8.705	Tootsville::Infinity-Get-Store-Item-Info .....	975
8.705.1	Function .....	975
8.705.2	Changes from 1.2 to 2.0 .....	975
8.705.3	200 OK .....	975
8.705.4	404 Not Found .....	975
8.705.5	File .....	975
8.706	Tootsville::Infinity-Get-User-Lists .....	976
8.706.1	Function .....	976
8.706.2	Changes from 1.2 to 2.0 .....	976
8.706.3	File .....	976
8.707	Tootsville::Infinity-Get-Wallet .....	977
8.707.1	Function .....	977
8.707.2	Changes from 1.1 to 1.2 .....	977
8.707.3	Changes from 1.2 to 2.0 .....	977
8.707.4	200 OK .....	977
8.707.5	File .....	977
8.708	Tootsville::Infinity-Get-Zone-List .....	978
8.708.1	Function .....	978
8.708.2	Changes from 1.2 to 2.0 .....	978
8.708.3	File .....	978
8.709	Tootsville::Infinity-Give .....	979
8.709.1	Function .....	979
8.709.2	412 Precondition Failed .....	979
8.709.3	404 Not Found .....	979
8.709.4	403 Forbidden .....	979
8.709.5	Changes from 1.2 to 2.0 .....	979
8.709.6	File .....	979
8.710	Tootsville::Infinity-Go .....	980
8.710.1	Function .....	980
8.710.2	Changes from 1.2 to 2.0 .....	980
8.710.3	File .....	980
8.711	Tootsville::Infinity-Init-User-Room .....	981
8.711.1	Function .....	981
8.711.2	410 Gone .....	981
8.711.3	File .....	981
8.712	Tootsville::Infinity-Join .....	982
8.712.1	Function .....	982
8.712.2	Usage .....	982
8.712.3	Status 200 OK .....	982
8.712.4	Error Return values .....	982
8.712.5	410 Gone .....	982
8.712.6	Changes from 1.2 to 2.0 .....	982
8.712.7	File .....	982
8.713	Tootsville::Infinity-Login .....	983
8.713.1	Function .....	983

8.713.2	Usage .....	983
8.713.3	Example .....	983
8.713.4	Changes from 1.2 to 2.0 .....	983
8.713.5	Changes from 1.1 to 1.2 .....	983
8.713.6	Changes from 1.0 to 1.1 .....	983
8.713.7	File .....	984
8.714	Tootsville::Infinity-Logout .....	985
8.714.1	Function .....	985
8.714.2	Changes from 1.2 to 2.0 .....	985
8.714.3	File .....	985
8.715	Tootsville::Infinity-Mail-Customer-Service .....	986
8.715.1	Function .....	986
8.715.2	Usage .....	986
8.715.3	File .....	986
8.716	Tootsville::Infinity-Peek-At-Inventory .....	987
8.716.1	Function .....	987
8.716.2	Usage .....	987
8.716.3	Examples .....	987
8.716.4	Status 200 OK .....	987
8.716.5	Status 404 Not Found .....	987
8.716.6	Status 400 Argument Error .....	987
8.716.7	File .....	987
8.717	Tootsville::Infinity-Ping .....	988
8.717.1	Function .....	988
8.717.2	Usage .....	988
8.717.3	Examples .....	988
8.717.4	200 OK .....	988
8.717.5	File .....	988
8.718	Tootsville::Infinity-Play-With .....	989
8.718.1	Function .....	989
8.718.2	Usage .....	989
8.718.3	Status 200 OK .....	989
8.718.4	Status 403 Not Your Toot .....	989
8.718.5	Status 404 No Such Toot .....	989
8.718.6	File .....	989
8.719	Tootsville::Infinity-Pre-Login .....	990
8.719.1	Function .....	990
8.719.2	Changes from 1.0 to 1.2 .....	990
8.719.3	Changes from 1.2 to 2.0 .....	990
8.719.4	File .....	990
8.720	Tootsville::Infinity-Prompt-Reply .....	991
8.720.1	Function .....	991
8.720.2	Usage .....	991
8.720.3	Overview of Prompts .....	991
8.720.4	Canceling a prompt .....	993
8.720.5	File .....	993
8.721	Tootsville::Infinity-Quiesce .....	994
8.721.1	Function .....	994

8.721.2	Usage	994
8.721.3	Status 200 OK	994
8.721.4	Asynchronous periodic demands	994
8.721.5	File	994
8.722	Tootsville::Infinity-Read-Map	995
8.722.1	Function	995
8.722.2	Usage	995
8.722.3	Status 200 OK	995
8.722.4	Overview of Spots and Badges	995
8.722.5	File	995
8.723	Tootsville::Infinity-Remove-From-List	996
8.723.1	Function	996
8.723.2	Usage	996
8.723.3	Status 200 OK	996
8.723.4	Status 404 Not Found	996
8.723.5	Status 412 Precondition Failed	996
8.723.6	File	996
8.724	Tootsville::Infinity-Report-Bug	997
8.724.1	Function	997
8.724.2	File	1001
8.725	Tootsville::Infinity-Report-User	1002
8.725.1	Function	1002
8.725.2	File	1002
8.726	Tootsville::Infinity-Request-Buddy	1003
8.726.1	Function	1003
8.726.2	Usage	1003
8.726.3	Example	1003
8.726.4	Changes from 1.0 to 1.1	1003
8.726.5	New in 1.1	1003
8.726.6	File	1003
8.727	Tootsville::Infinity-Send-Mail-Message	1004
8.727.1	Function	1004
8.727.2	Usage	1004
8.727.3	Examples	1004
8.727.4	Changes from 1.2 to 2.0	1004
8.727.5	Formerly Proprietary Extension	1004
8.727.6	200 OK	1004
8.727.7	400 Bad Request	1005
8.727.8	413 Payload Too Large	1005
8.727.9	Formerly Proprietary Extension	1005
8.727.10	File	1005
8.728	Tootsville::Infinity-Send-Out-Of-Band-Message	1006
8.728.1	Function	1006
8.728.2	File	1006
8.729	Tootsville::Infinity-Server-Time	1007
8.729.1	Function	1007
8.729.2	Usage	1007
8.729.3	Example	1007

8.729.4	File .....	1007
8.730	Tootsville::Infinity-Set-Avatar-Color .....	1008
8.730.1	Function .....	1008
8.730.2	Romance 1.1 instructions .....	1008
8.730.3	File .....	1008
8.731	Tootsville::Infinity-Set-Furniture .....	1009
8.731.1	Function .....	1009
8.731.2	Romance 1.2 instructions .....	1009
8.731.3	Changes from 1.2 to 2.0 .....	1010
8.731.4	200 OK .....	1010
8.731.5	400 Error in parameters .....	1010
8.731.6	File .....	1010
8.732	Tootsville::Infinity-Set-Room-Var .....	1011
8.732.1	Function .....	1011
8.732.2	Usage .....	1011
8.732.3	Example .....	1011
8.732.4	Changes from 1.2 to 2.0 .....	1011
8.732.5	File .....	1011
8.733	Tootsville::Infinity-Set-User-Var .....	1012
8.733.1	Function .....	1012
8.733.2	Usage .....	1012
8.733.3	Example .....	1012
8.733.4	Changes from 1.2 to 2.0 .....	1012
8.733.5	Available Attributes (2.0) .....	1012
8.733.6	200 OK .....	1013
8.733.7	400 Illegal .....	1013
8.733.8	File .....	1013
8.734	Tootsville::Infinity-Shoot .....	1014
8.734.1	Function .....	1014
8.734.2	Usage .....	1014
8.734.3	Example .....	1014
8.734.4	See also .....	1014
8.734.5	File .....	1014
8.735	Tootsville::Infinity-Spawn-Zone .....	1015
8.735.1	Function .....	1015
8.735.2	Implementation in 2.0 .....	1015
8.735.3	Changes from 1.2 to 2.0 .....	1015
8.735.4	File .....	1015
8.736	Tootsville::Infinity-Speak .....	1016
8.736.1	Function .....	1016
8.736.2	Usage .....	1016
8.736.3	Speech filtering .....	1016
8.736.4	Special character handling .....	1016
8.736.5	Special commands .....	1018
8.736.6	Changes from 1.2 to 2.0 .....	1018
8.736.7	File .....	1018
8.737	Tootsville::Infinity-Stamp-Passport .....	1019
8.737.1	Function .....	1019

8.737.2	Usage	1019
8.737.3	Example	1019
8.737.4	Changes from 1.2 to 2.0	1019
8.737.5	Formerly Proprietary Extension	1019
8.737.6	File	1019
8.738	Tootsville::Infinity-Start-Event	1020
8.738.1	Function	1020
8.738.2	What is a “Quaestor Event”?	1020
8.738.3	Usage	1020
8.738.4	Responses	1020
8.738.4.1	Event already completed	1020
8.738.4.2	Event started successfully	1021
8.738.4.3	Event requires a download to begin	1021
8.738.5	Error response	1022
8.738.6	Ending an event	1022
8.738.7	Quaestor Events in Detail	1022
8.738.7.1	Magic Fountains	1022
8.738.7.2	Shops	1022
8.738.7.3	Secrets and Treasures	1022
8.738.7.4	Minigames	1022
8.738.8	Changes from 1.2 to 2.0	1022
8.738.9	File	1022
8.739	Tootsville::Infinity-Stats	1023
8.739.1	Function	1023
8.739.2	File	1023
8.740	Tootsville::Infinity-Toot-List	1024
8.740.1	Function	1024
8.740.2	Usage	1024
8.740.3	200 OK	1024
8.740.4	File	1024
8.741	Tootsville::Infinity-Use-Equipment	1025
8.741.1	Function	1025
8.741.2	Usage	1025
8.741.3	Changes from 1.2 to 2.0	1025
8.741.4	File	1025
8.742	Tootsville::Infinity-Wardrobe	1026
8.742.1	Function	1026
8.742.2	Usage	1026
8.742.3	200 OK	1026
8.742.4	Changes from 1.2 to 2.0	1026
8.742.5	File	1026
8.743	Tootsville::Infinity-Websocket-Resource	1027
8.743.1	Class	1027
8.743.2	Slots	1027
8.744	Tootsville::Infinity-Wtl	1028
8.744.1	Function	1028
8.744.2	Usage	1028
8.744.3	Reply	1028

8.744.4	Future Directions .....	1028
8.744.5	See Also .....	1028
8.744.6	Changes from 1.1 .....	1028
8.744.7	File .....	1029
8.745	Tootsville::Infinity-Wtl-4 .....	1030
8.745.1	Function .....	1030
8.745.2	Usage .....	1030
8.745.3	File .....	1030
8.746	Tootsville::Init-Async .....	1031
8.746.1	Function .....	1031
8.746.2	File .....	1031
8.747	Tootsville::Init-Characters .....	1032
8.747.1	Function .....	1032
8.747.2	File .....	1032
8.748	Tootsville::Integer-To-Byte-Vector .....	1033
8.748.1	Function .....	1033
8.748.2	File .....	1033
8.749	Tootsville::Integer-To-Color24 .....	1034
8.749.1	Function .....	1034
8.749.2	File .....	1034
8.750	Tootsville::Interpret-Facing .....	1035
8.750.1	Function .....	1035
8.750.2	Changes from 1.2 to 2.0 .....	1035
8.750.3	File .....	1035
8.751	Tootsville::Invalidate-Cache .....	1036
8.751.1	Function .....	1036
8.751.2	File .....	1036
8.752	Tootsville::Inventory-Item .....	1037
8.752.1	Class .....	1037
8.752.2	Slots .....	1037
8.753	Tootsville::Inventory-Item-Equipped .....	1038
8.753.1	Function .....	1038
8.753.2	File .....	1038
8.753.3	SetF Function .....	1038
8.753.4	File .....	1038
8.754	Tootsville::Inventory-Item-Equipped-P .....	1039
8.754.1	Function .....	1039
8.754.2	File .....	1039
8.755	Tootsville::Inventory-Item-Item .....	1040
8.755.1	Function .....	1040
8.755.2	File .....	1040
8.755.3	SetF Function .....	1040
8.755.4	File .....	1040
8.756	Tootsville::Inventory-Item-P .....	1041
8.756.1	Function .....	1041
8.756.2	File .....	1041
8.757	Tootsville::Inventory-Item-Person .....	1042
8.757.1	Function .....	1042



8.757.2	File	1042
8.757.3	SetF Function	1042
8.757.4	File	1042
8.758	Tootsville::Inventory-Item-Toot	1043
8.758.1	Function	1043
8.758.2	File	1043
8.758.3	SetF Function	1043
8.758.4	File	1043
8.759	Tootsville::Item	1044
8.759.1	Class	1044
8.759.2	Slots	1044
8.760	Tootsville::Item-Alt-Color	1045
8.760.1	Function	1045
8.760.2	File	1045
8.760.3	SetF Function	1045
8.760.4	File	1045
8.761	Tootsville::Item-Altitude	1046
8.761.1	Function	1046
8.761.2	File	1046
8.761.3	SetF Function	1046
8.761.4	File	1046
8.762	Tootsville::Item-Avatar-Scale-X	1047
8.762.1	Function	1047
8.762.2	File	1047
8.762.3	SetF Function	1047
8.762.4	File	1047
8.763	Tootsville::Item-Avatar-Scale-Y	1048
8.763.1	Function	1048
8.763.2	File	1048
8.763.3	SetF Function	1048
8.763.4	File	1048
8.764	Tootsville::Item-Avatar-Scale-Z	1049
8.764.1	Function	1049
8.764.2	File	1049
8.764.3	SetF Function	1049
8.764.4	File	1049
8.765	Tootsville::Item-Base-Color	1050
8.765.1	Function	1050
8.765.2	File	1050
8.765.3	SetF Function	1050
8.765.4	File	1050
8.766	Tootsville::Item-Energy	1051
8.766.1	Function	1051
8.766.2	File	1051
8.766.3	SetF Function	1051
8.766.4	File	1051
8.767	Tootsville::Item-Facing	1052
8.767.1	Function	1052

8.767.2	File	1052
8.767.3	SetF Function	1052
8.767.4	File	1052
8.768	Tootsville::Item-Gain-Energy	1053
8.768.1	Function	1053
8.768.2	File	1053
8.769	Tootsville::Item-In-Inventory-P	1054
8.769.1	Function	1054
8.769.2	File	1054
8.770	Tootsville::Item-Info	1055
8.770.1	Function	1055
8.770.2	File	1055
8.771	Tootsville::Item-Latitude	1056
8.771.1	Function	1056
8.771.2	File	1056
8.771.3	SetF Function	1056
8.771.4	File	1056
8.772	Tootsville::Item-Longitude	1057
8.772.1	Function	1057
8.772.2	File	1057
8.772.3	SetF Function	1057
8.772.4	File	1057
8.773	Tootsville::Item-Lose-Energy	1058
8.773.1	Function	1058
8.773.2	File	1058
8.774	Tootsville::Item-Owned-By-P	1059
8.774.1	Function	1059
8.774.2	File	1059
8.775	Tootsville::Item-P	1060
8.775.1	Function	1060
8.775.2	File	1060
8.776	Tootsville::Item-Template	1061
8.776.1	Function	1061
8.776.2	File	1061
8.776.3	SetF Function	1061
8.776.4	File	1061
8.776.5	Class	1061
8.776.6	Slots	1061
8.777	Tootsville::Item-Template-Avatar	1062
8.777.1	Function	1062
8.777.2	File	1062
8.777.3	SetF Function	1062
8.777.4	File	1062
8.778	Tootsville::Item-Template-Avatar-Scale-X	1063
8.778.1	Function	1063
8.778.2	File	1063
8.778.3	SetF Function	1063
8.778.4	File	1063

8.779	Tootsville::Item-Template-Avatar-Scale-Y .....	1064
8.779.1	Function .....	1064
8.779.2	File .....	1064
8.779.3	SetF Function .....	1064
8.779.4	File .....	1064
8.780	Tootsville::Item-Template-Avatar-Scale-Z .....	1065
8.780.1	Function .....	1065
8.780.2	File .....	1065
8.780.3	SetF Function .....	1065
8.780.4	File .....	1065
8.781	Tootsville::Item-Template-Default-Alt-Color .....	1066
8.781.1	Function .....	1066
8.781.2	File .....	1066
8.781.3	SetF Function .....	1066
8.781.4	File .....	1066
8.782	Tootsville::Item-Template-Default-Base-Color .....	1067
8.782.1	Function .....	1067
8.782.2	File .....	1067
8.782.3	SetF Function .....	1067
8.782.4	File .....	1067
8.783	Tootsville::Item-Template-Description .....	1068
8.783.1	Function .....	1068
8.783.2	File .....	1068
8.783.3	SetF Function .....	1068
8.783.4	File .....	1068
8.784	Tootsville::Item-Template-Energy-Kind .....	1069
8.784.1	Function .....	1069
8.784.2	File .....	1069
8.784.3	SetF Function .....	1069
8.784.4	File .....	1069
8.785	Tootsville::Item-Template-Energy-Max .....	1070
8.785.1	Function .....	1070
8.785.2	File .....	1070
8.785.3	SetF Function .....	1070
8.785.4	File .....	1070
8.786	Tootsville::Item-Template-Id .....	1071
8.786.1	Function .....	1071
8.786.2	File .....	1071
8.786.3	SetF Function .....	1071
8.786.4	File .....	1071
8.787	Tootsville::Item-Template-Info .....	1072
8.787.1	Function .....	1072
8.787.2	File .....	1072
8.788	Tootsville::Item-Template-Name .....	1073
8.788.1	Function .....	1073
8.788.2	File .....	1073
8.788.3	SetF Function .....	1073
8.788.4	File .....	1073

8.789	Tootsville::Item-Template-On-Zero .....	1074
8.789.1	Function .....	1074
8.789.2	File .....	1074
8.789.3	SetF Function .....	1074
8.789.4	File .....	1074
8.790	Tootsville::Item-Template-P .....	1075
8.790.1	Function .....	1075
8.790.2	File .....	1075
8.791	Tootsville::Item-Template-Trade .....	1076
8.791.1	Function .....	1076
8.791.2	File .....	1076
8.791.3	SetF Function .....	1076
8.791.4	File .....	1076
8.792	Tootsville::Item-Template-Wear-Slot .....	1077
8.792.1	Function .....	1077
8.792.2	File .....	1077
8.792.3	SetF Function .....	1077
8.792.4	File .....	1077
8.793	Tootsville::Item-Template-Weight .....	1078
8.793.1	Function .....	1078
8.793.2	File .....	1078
8.793.3	SetF Function .....	1078
8.793.4	File .....	1078
8.794	Tootsville::Item-Uuid .....	1079
8.794.1	Function .....	1079
8.794.2	File .....	1079
8.794.3	SetF Function .....	1079
8.794.4	File .....	1079
8.795	Tootsville::Item-World .....	1080
8.795.1	Function .....	1080
8.795.2	File .....	1080
8.795.3	SetF Function .....	1080
8.795.4	File .....	1080
8.796	Tootsville::Item-X .....	1081
8.796.1	Function .....	1081
8.796.2	File .....	1081
8.796.3	SetF Function .....	1081
8.796.4	File .....	1081
8.797	Tootsville::Item-Y .....	1082
8.797.1	Function .....	1082
8.797.2	File .....	1082
8.797.3	SetF Function .....	1082
8.797.4	File .....	1082
8.798	Tootsville::Item-Z .....	1083
8.798.1	Function .....	1083
8.798.2	File .....	1083
8.798.3	SetF Function .....	1083
8.798.4	File .....	1083

8.799	Tootsville::Items-At	1084
8.799.1	Function	1084
8.799.2	File	1084
8.800	Tootsville::Jack-Personality	1085
8.800.1	Class	1085
8.800.2	Slots	1085
8.801	Tootsville::Journal	1086
8.801.1	Function	1086
8.801.2	Usage	1086
8.801.3	Examples	1086
8.801.4	File	1086
8.802	Tootsville::Json-To-Html	1087
8.802.1	Function	1087
8.802.2	File	1087
8.803	Tootsville::Kick	1088
8.803.1	Function	1088
8.803.2	File	1088
8.804	Tootsville::Kick-Child-Time-Up	1089
8.804.1	Function	1089
8.804.2	File	1089
8.805	Tootsville::Kind-Of-Habitat	1090
8.805.1	Type	1090
8.806	Tootsville::Lambda-List-As-Variables	1091
8.806.1	Function	1091
8.806.2	File	1091
8.807	Tootsville::Last-Active	1092
8.807.1	Function	1092
8.807.2	SetF Function	1092
8.808	Tootsville::Latitude	1093
8.808.1	Function	1093
8.808.2	File	1093
8.809	Tootsville::Legal-Age	1094
8.809.1	Function	1094
8.809.2	File	1094
8.810	Tootsville::Lil-Mc-Personality	1095
8.810.1	Class	1095
8.810.2	Slots	1095
8.811	Tootsville::Limit-String-Length	1096
8.811.1	Function	1096
8.811.2	File	1096
8.812	Tootsville::Lisp-To-Db-Name	1097
8.812.1	Function	1097
8.812.2	File	1097
8.813	Tootsville::List-Banhammers	1098
8.813.1	Function	1098
8.813.2	File	1098
8.814	Tootsville::List-Of-String=	1099
8.814.1	Function	1099

8.814.2	File	1099
8.815	Tootsville::Listen-For-Websockets	1100
8.815.1	Function	1100
8.815.2	File	1100
8.816	Tootsville::Listener-Name	1101
8.816.1	Function	1101
8.816.2	File	1101
8.817	Tootsville::Load-Config	1102
8.817.1	Function	1102
8.817.2	File	1102
8.818	Tootsville::Load-Record	1103
8.818.1	Function	1103
8.818.2	File	1103
8.819	Tootsville::Local-Room-Vars	1104
8.819.1	Function	1104
8.819.2	File	1104
8.820	Tootsville::Locale-Music	1105
8.820.1	Class	1105
8.820.2	Slots	1105
8.821	Tootsville::Locale-Music-Music	1106
8.821.1	Function	1106
8.821.2	File	1106
8.821.3	SetF Function	1106
8.821.4	File	1106
8.822	Tootsville::Locale-Music-P	1107
8.822.1	Function	1107
8.822.2	File	1107
8.823	Tootsville::Locale-Music-Radius	1108
8.823.1	Function	1108
8.823.2	File	1108
8.823.3	SetF Function	1108
8.823.4	File	1108
8.824	Tootsville::Locale-Music-X	1109
8.824.1	Function	1109
8.824.2	File	1109
8.824.3	SetF Function	1109
8.824.4	File	1109
8.825	Tootsville::Locale-Music-Y	1110
8.825.1	Function	1110
8.825.2	File	1110
8.825.3	SetF Function	1110
8.825.4	File	1110
8.826	Tootsville::Locale-Music-Z	1111
8.826.1	Function	1111
8.826.2	File	1111
8.826.3	SetF Function	1111
8.826.4	File	1111
8.827	Tootsville::Login	1112

8.827.1	Class .....	1112
8.827.2	Slots .....	1112
8.828	Tootsville::Login-Child .....	1113
8.828.1	Function .....	1113
8.828.2	File .....	1113
8.829	Tootsville::Login-Credential .....	1114
8.829.1	Function .....	1114
8.829.2	File .....	1114
8.829.3	SetF Function .....	1114
8.829.4	File .....	1114
8.830	Tootsville::Login-Fail .....	1115
8.830.1	Function .....	1115
8.830.2	File .....	1115
8.831	Tootsville::Login-Failed-Message .....	1116
8.831.1	Function .....	1116
8.831.2	File .....	1116
8.832	Tootsville::Login-Last-Seen .....	1117
8.832.1	Function .....	1117
8.832.2	File .....	1117
8.832.3	SetF Function .....	1117
8.832.4	File .....	1117
8.833	Tootsville::Login-Ok-Message .....	1118
8.833.1	Function .....	1118
8.833.2	File .....	1118
8.834	Tootsville::Login-Origin .....	1119
8.834.1	Function .....	1119
8.834.2	File .....	1119
8.834.3	SetF Function .....	1119
8.834.4	File .....	1119
8.835	Tootsville::Login-P .....	1120
8.835.1	Function .....	1120
8.835.2	File .....	1120
8.836	Tootsville::Login-Person .....	1121
8.836.1	Function .....	1121
8.836.2	File .....	1121
8.836.3	SetF Function .....	1121
8.836.4	File .....	1121
8.837	Tootsville::Login-Renewed .....	1122
8.837.1	Function .....	1122
8.837.2	File .....	1122
8.837.3	SetF Function .....	1122
8.837.4	File .....	1122
8.838	Tootsville::Login-Start .....	1123
8.838.1	Function .....	1123
8.838.2	File .....	1123
8.838.3	SetF Function .....	1123
8.838.4	File .....	1123
8.839	Tootsville::Login-Uuid .....	1124

8.839.1	Function	1124
8.839.2	File	1124
8.839.3	SetF Function	1124
8.839.4	File	1124
8.840	Tootsville::Longitude	1125
8.840.1	Function	1125
8.840.2	File	1125
8.841	Tootsville::Look-For-Ssl-Certs	1126
8.841.1	Function	1126
8.841.2	File	1126
8.842	Tootsville::Lot	1127
8.842.1	Class	1127
8.842.2	Slots	1127
8.843	Tootsville::Lot-Owner-Toot	1128
8.843.1	Function	1128
8.843.2	File	1128
8.843.3	SetF Function	1128
8.843.4	File	1128
8.844	Tootsville::Lot-Ownership	1129
8.844.1	Function	1129
8.844.2	File	1129
8.844.3	SetF Function	1129
8.844.4	File	1129
8.845	Tootsville::Lot-P	1130
8.845.1	Function	1130
8.845.2	File	1130
8.846	Tootsville::Lot-World	1131
8.846.1	Function	1131
8.846.2	File	1131
8.846.3	SetF Function	1131
8.846.4	File	1131
8.847	Tootsville::Lot-X1	1132
8.847.1	Function	1132
8.847.2	File	1132
8.847.3	SetF Function	1132
8.847.4	File	1132
8.848	Tootsville::Lot-X2	1133
8.848.1	Function	1133
8.848.2	File	1133
8.848.3	SetF Function	1133
8.848.4	File	1133
8.849	Tootsville::Lot-Y1	1134
8.849.1	Function	1134
8.849.2	File	1134
8.849.3	SetF Function	1134
8.849.4	File	1134
8.850	Tootsville::Lot-Y2	1135
8.850.1	Function	1135



8.850.2	File .....	1135
8.850.3	SetF Function .....	1135
8.850.4	File .....	1135
8.851	Tootsville::Lot-Z1 .....	1136
8.851.1	Function .....	1136
8.851.2	File .....	1136
8.851.3	SetF Function .....	1136
8.851.4	File .....	1136
8.852	Tootsville::Lot-Z2 .....	1137
8.852.1	Function .....	1137
8.852.2	File .....	1137
8.852.3	SetF Function .....	1137
8.852.4	File .....	1137
8.853	Tootsville::Make-Avatar .....	1138
8.853.1	Function .....	1138
8.853.2	File .....	1138
8.854	Tootsville::Make-Avatar-Slot .....	1139
8.854.1	Function .....	1139
8.854.2	File .....	1139
8.855	Tootsville::Make-Character-Music .....	1140
8.855.1	Function .....	1140
8.855.2	File .....	1140
8.856	Tootsville::Make-Child-Request .....	1141
8.856.1	Function .....	1141
8.856.2	File .....	1141
8.857	Tootsville::Make-Color24 .....	1142
8.857.1	Function .....	1142
8.857.2	File .....	1142
8.858	Tootsville::Make-Contact .....	1143
8.858.1	Function .....	1143
8.858.2	File .....	1143
8.859	Tootsville::Make-Credential .....	1144
8.859.1	Function .....	1144
8.859.2	File .....	1144
8.860	Tootsville::Make-Endpoint-Function-Name .....	1145
8.860.1	Function .....	1145
8.860.2	File .....	1145
8.861	Tootsville::Make-Game-Point .....	1146
8.861.1	Function .....	1146
8.861.2	File .....	1146
8.862	Tootsville::Make-Gossip-Initiation .....	1147
8.862.1	Function .....	1147
8.862.2	File .....	1147
8.863	Tootsville::Make-Inventory-Item .....	1148
8.863.1	Function .....	1148
8.863.2	File .....	1148
8.864	Tootsville::Make-Item .....	1149
8.864.1	Function .....	1149

8.864.2	File	1149
8.865	Tootsville::Make-Item-Template	1150
8.865.1	Function	1150
8.865.2	File	1150
8.866	Tootsville::Make-Locale-Music	1151
8.866.1	Function	1151
8.866.2	File	1151
8.867	Tootsville::Make-Login	1152
8.867.1	Function	1152
8.867.2	File	1152
8.868	Tootsville::Make-Lot	1153
8.868.1	Function	1153
8.868.2	File	1153
8.869	Tootsville::Make-Metronome-Task	1154
8.869.1	Function	1154
8.869.2	File	1154
8.870	Tootsville::Make-Mist	1155
8.870.1	Function	1155
8.870.2	File	1155
8.871	Tootsville::Make-Music	1156
8.871.1	Function	1156
8.871.2	File	1156
8.872	Tootsville::Make-New-Toot-State	1157
8.872.1	Function	1157
8.872.2	File	1157
8.873	Tootsville::Make-Parent-Child	1158
8.873.1	Function	1158
8.873.2	File	1158
8.874	Tootsville::Make-Pattern	1159
8.874.1	Function	1159
8.874.2	File	1159
8.875	Tootsville::Make-Person	1160
8.875.1	Function	1160
8.875.2	File	1160
8.876	Tootsville::Make-Person-Link	1161
8.876.1	Function	1161
8.876.2	File	1161
8.877	Tootsville::Make-Place	1162
8.877.1	Function	1162
8.877.2	File	1162
8.878	Tootsville::Make-Quaestor-Event	1163
8.878.1	Function	1163
8.878.2	File	1163
8.879	Tootsville::Make-Record	1164
8.879.1	Function	1164
8.879.2	File	1164
8.880	Tootsville::Make-Sms	1165
8.880.1	Function	1165

8.880.2	File	1165
8.881	Tootsville::Make-Store-Item	1166
8.881.1	Function	1166
8.881.2	File	1166
8.882	Tootsville::Make-Tcp-Client	1167
8.882.1	Function	1167
8.882.2	File	1167
8.883	Tootsville::Make-Terrain-Height	1168
8.883.1	Function	1168
8.883.2	File	1168
8.884	Tootsville::Make-Toot	1169
8.884.1	Function	1169
8.884.2	File	1169
8.885	Tootsville::Make-Toot-Quiesced	1170
8.885.1	Function	1170
8.885.2	File	1170
8.886	Tootsville::Make-Wear-Slot	1171
8.886.1	Function	1171
8.886.2	File	1171
8.887	Tootsville::Make-Wind-Vector	1172
8.887.1	Function	1172
8.887.2	File	1172
8.888	Tootsville::Make-Wind-Vector-Field	1173
8.888.1	Function	1173
8.888.2	File	1173
8.889	Tootsville::Make-World	1174
8.889.1	Function	1174
8.889.2	File	1174
8.890	Tootsville::Make-Wtl-Course	1175
8.890.1	Function	1175
8.890.2	File	1175
8.891	Tootsville::Map-Places	1176
8.891.1	Type	1176
8.892	Tootsville::Maybe-Parent-Approval	1177
8.892.1	Function	1177
8.892.2	File	1177
8.893	Tootsville::Mayor-Louis-Personality	1178
8.893.1	Class	1178
8.893.2	Slots	1178
8.894	Tootsville::Memcached-Get-Key	1179
8.894.1	Function	1179
8.894.2	File	1179
8.895	Tootsville::Metronome-Idle-Tasks	1180
8.895.1	Function	1180
8.895.2	File	1180
8.896	Tootsville::Metronome-Register	1181
8.896.1	Function	1181
8.896.2	File	1181

8.897	Tootsville::Metronome-Remove	1182
8.897.1	Function	1182
8.897.2	File	1182
8.898	Tootsville::Metronome-Task	1183
8.898.1	Class	1183
8.898.2	Slots	1183
8.899	Tootsville::Metronome-Task-Frequency	1184
8.899.1	Function	1184
8.899.2	File	1184
8.899.3	SetF Function	1184
8.899.4	File	1184
8.900	Tootsville::Metronome-Task-Function	1185
8.900.1	Function	1185
8.900.2	File	1185
8.900.3	SetF Function	1185
8.900.4	File	1185
8.901	Tootsville::Metronome-Task-Name	1186
8.901.1	Function	1186
8.901.2	File	1186
8.901.3	SetF Function	1186
8.901.4	File	1186
8.902	Tootsville::Metronome-Task-One-Shot-Time	1187
8.902.1	Function	1187
8.902.2	File	1187
8.902.3	SetF Function	1187
8.902.4	File	1187
8.903	Tootsville::Metronome-Task-P	1188
8.903.1	Function	1188
8.903.2	File	1188
8.904	Tootsville::Metronome-Task-Thread	1189
8.904.1	Function	1189
8.904.2	File	1189
8.904.3	SetF Function	1189
8.904.4	File	1189
8.905	Tootsville::Mist	1190
8.905.1	Class	1190
8.905.2	Slots	1190
8.906	Tootsville::Mist-Altitude-1	1191
8.906.1	Function	1191
8.906.2	File	1191
8.906.3	SetF Function	1191
8.906.4	File	1191
8.907	Tootsville::Mist-Altitude-2	1192
8.907.1	Function	1192
8.907.2	File	1192
8.907.3	SetF Function	1192
8.907.4	File	1192
8.908	Tootsville::Mist-Definedp	1193

8.908.1	Function	1193
8.908.2	File	1193
8.908.3	SetF Function	1193
8.908.4	File	1193
8.909	Tootsville::Mist-Latitude-1	1194
8.909.1	Function	1194
8.909.2	File	1194
8.909.3	SetF Function	1194
8.909.4	File	1194
8.910	Tootsville::Mist-Latitude-2	1195
8.910.1	Function	1195
8.910.2	File	1195
8.910.3	SetF Function	1195
8.910.4	File	1195
8.911	Tootsville::Mist-Longitude-1	1196
8.911.1	Function	1196
8.911.2	File	1196
8.911.3	SetF Function	1196
8.911.4	File	1196
8.912	Tootsville::Mist-Longitude-2	1197
8.912.1	Function	1197
8.912.2	File	1197
8.912.3	SetF Function	1197
8.912.4	File	1197
8.913	Tootsville::Mist-P	1198
8.913.1	Function	1198
8.913.2	File	1198
8.914	Tootsville::Mist-World	1199
8.914.1	Function	1199
8.914.2	File	1199
8.914.3	SetF Function	1199
8.914.4	File	1199
8.915	Tootsville::Moo-Personality	1200
8.915.1	Class	1200
8.915.2	Slots	1200
8.916	Tootsville::Moon-Position	1201
8.916.1	Function	1201
8.916.2	File	1201
8.917	Tootsville::Music	1202
8.917.1	Class	1202
8.917.2	Slots	1202
8.918	Tootsville::Music-Artist	1203
8.918.1	Function	1203
8.918.2	File	1203
8.918.3	SetF Function	1203
8.918.4	File	1203
8.919	Tootsville::Music-Genre	1204
8.919.1	Function	1204

8.919.2	File	1204
8.919.3	SetF Function	1204
8.919.4	File	1204
8.920	Tootsville::Music-Id	1205
8.920.1	Function	1205
8.920.2	File	1205
8.920.3	SetF Function	1205
8.920.4	File	1205
8.921	Tootsville::Music-License	1206
8.921.1	Function	1206
8.921.2	File	1206
8.921.3	SetF Function	1206
8.921.4	File	1206
8.922	Tootsville::Music-Moniker	1207
8.922.1	Function	1207
8.922.2	File	1207
8.922.3	SetF Function	1207
8.922.4	File	1207
8.923	Tootsville::Music-P	1208
8.923.1	Function	1208
8.923.2	File	1208
8.924	Tootsville::Music-Title	1209
8.924.1	Function	1209
8.924.2	File	1209
8.924.3	SetF Function	1209
8.924.4	File	1209
8.925	Tootsville::Name-For-Content-Type	1210
8.925.1	Function	1210
8.925.2	File	1210
8.926	Tootsville::Name-Idle-Threads-Sequentially	1211
8.926.1	Function	1211
8.926.2	File	1211
8.927	Tootsville::Nearp	1212
8.927.1	Function	1212
8.927.2	File	1212
8.928	Tootsville::Nevermind-Personality	1213
8.928.1	Class	1213
8.928.2	Slots	1213
8.929	Tootsville::Normalize-Url	1214
8.929.1	Function	1214
8.929.2	File	1214
8.930	Tootsville::Not-Found	1215
8.930.1	Class	1215
8.930.2	Slots	1215
8.931	Tootsville::Not-Found-If-Null	1216
8.931.1	Function	1216
8.931.2	File	1216
8.932	Tootsville::Not-Found-Thing	1217

8.932.1	Function	1217
8.932.2	SetF Function	1217
8.933	Tootsville::Not-Your-Toot-Error	1218
8.933.1	Class	1218
8.933.2	Slots	1218
8.934	Tootsville::Null-If-Empty	1219
8.934.1	Function	1219
8.934.2	File	1219
8.935	Tootsville::On-Exception	1220
8.935.1	Function	1220
8.936	Tootsville::Open-Log-File	1221
8.936.1	Function	1221
8.936.2	File	1221
8.937	Tootsville::Pad-To-Multiple-Of-8	1222
8.937.1	Function	1222
8.937.2	File	1222
8.938	Tootsville::Parent-Child	1223
8.938.1	Class	1223
8.938.2	Slots	1223
8.939	Tootsville::Parent-Child-Child	1224
8.939.1	Function	1224
8.939.2	File	1224
8.939.3	SetF Function	1224
8.939.4	File	1224
8.940	Tootsville::Parent-Child-P	1225
8.940.1	Function	1225
8.940.2	File	1225
8.941	Tootsville::Parent-Child-Parent	1226
8.941.1	Function	1226
8.941.2	File	1226
8.941.3	SetF Function	1226
8.941.4	File	1226
8.942	Tootsville::Parent-Deny-Permission	1227
8.942.1	Function	1227
8.942.2	File	1227
8.943	Tootsville::Parent-Grant-Permission	1228
8.943.1	Function	1228
8.943.2	File	1228
8.944	Tootsville::Parse-Backtrace	1229
8.944.1	Function	1229
8.944.2	File	1229
8.945	Tootsville::Parse-Color24	1230
8.945.1	Function	1230
8.945.2	File	1230
8.946	Tootsville::Parse-Operator-Command	1231
8.946.1	Function	1231
8.946.2	File	1231
8.947	Tootsville::Parse-Uri-As-Template	1232

8.947.1	Function	1232
8.947.2	File	1232
8.948	Tootsville::Parse-Wtl-For-Robot	1233
8.948.1	Function	1233
8.948.2	File	1233
8.949	Tootsville::Path->Openapi	1234
8.949.1	Function	1234
8.949.2	File	1234
8.950	Tootsville::Pattern	1235
8.950.1	Class	1235
8.950.2	Slots	1235
8.951	Tootsville::Pattern-Id	1236
8.951.1	Function	1236
8.951.2	File	1236
8.951.3	SetF Function	1236
8.951.4	File	1236
8.952	Tootsville::Pattern-Name	1237
8.952.1	Function	1237
8.952.2	File	1237
8.952.3	SetF Function	1237
8.952.4	File	1237
8.953	Tootsville::Pattern-P	1238
8.953.1	Function	1238
8.953.2	File	1238
8.954	Tootsville::Peer-Address	1239
8.954.1	Function	1239
8.955	Tootsville::Pending-Child-Approval-Request	1240
8.955.1	Function	1240
8.955.2	File	1240
8.956	Tootsville::Pending-Child-Requests-By-Toot	1241
8.956.1	Function	1241
8.956.2	File	1241
8.957	Tootsville::Person	1242
8.957.1	Class	1242
8.957.2	Slots	1242
8.958	Tootsville::Person-Age	1243
8.958.1	Function	1243
8.958.2	File	1243
8.958.3	SetF Function	1243
8.958.4	File	1243
8.959	Tootsville::Person-Age*	1244
8.959.1	Function	1244
8.959.2	File	1244
8.960	Tootsville::Person-Date-Of-Birth	1245
8.960.1	Function	1245
8.960.2	File	1245
8.960.3	SetF Function	1245
8.960.4	File	1245



8.961	Tootsville::Person-Display-Name	1246
8.961.1	Function	1246
8.961.2	File	1246
8.961.3	SetF Function	1246
8.961.4	File	1246
8.962	Tootsville::Person-First-Email	1247
8.962.1	Function	1247
8.962.2	File	1247
8.963	Tootsville::Person-Gender	1248
8.963.1	Function	1248
8.963.2	File	1248
8.963.3	SetF Function	1248
8.963.4	File	1248
8.964	Tootsville::Person-Given-Name	1249
8.964.1	Function	1249
8.964.2	File	1249
8.964.3	SetF Function	1249
8.964.4	File	1249
8.965	Tootsville::Person-Info	1250
8.965.1	Function	1250
8.965.2	File	1250
8.966	Tootsville::Person-Is-Patron-P	1251
8.966.1	Function	1251
8.966.2	File	1251
8.967	Tootsville::Person-Lang	1252
8.967.1	Function	1252
8.967.2	File	1252
8.967.3	SetF Function	1252
8.967.4	File	1252
8.968	Tootsville::Person-Link	1253
8.968.1	Class	1253
8.968.2	Slots	1253
8.969	Tootsville::Person-Link-Label	1254
8.969.1	Function	1254
8.969.2	File	1254
8.969.3	SetF Function	1254
8.969.4	File	1254
8.970	Tootsville::Person-Link-P	1255
8.970.1	Function	1255
8.970.2	File	1255
8.971	Tootsville::Person-Link-Person	1256
8.971.1	Function	1256
8.971.2	File	1256
8.971.3	SetF Function	1256
8.971.4	File	1256
8.972	Tootsville::Person-Link-Provenance	1257
8.972.1	Function	1257
8.972.2	File	1257

8.972.3	SetF Function .....	1257
8.972.4	File .....	1257
8.973	Tootsville::Person-Link-Rel .....	1258
8.973.1	Function .....	1258
8.973.2	File .....	1258
8.973.3	SetF Function .....	1258
8.973.4	File .....	1258
8.974	Tootsville::Person-Link-Url .....	1259
8.974.1	Function .....	1259
8.974.2	File .....	1259
8.974.3	SetF Function .....	1259
8.974.4	File .....	1259
8.975	Tootsville::Person-Link-Uuid .....	1260
8.975.1	Function .....	1260
8.975.2	File .....	1260
8.975.3	SetF Function .....	1260
8.975.4	File .....	1260
8.976	Tootsville::Person-Links-To-Email .....	1261
8.976.1	Function .....	1261
8.976.2	File .....	1261
8.977	Tootsville::Person-P .....	1262
8.977.1	Function .....	1262
8.977.2	File .....	1262
8.978	Tootsville::Person-Sensitivep .....	1263
8.978.1	Function .....	1263
8.978.2	File .....	1263
8.978.3	SetF Function .....	1263
8.978.4	File .....	1263
8.979	Tootsville::Person-Surname .....	1264
8.979.1	Function .....	1264
8.979.2	File .....	1264
8.979.3	SetF Function .....	1264
8.979.4	File .....	1264
8.980	Tootsville::Person-Uuid .....	1265
8.980.1	Function .....	1265
8.980.2	File .....	1265
8.980.3	SetF Function .....	1265
8.980.4	File .....	1265
8.981	Tootsville::Picasso-Personality .....	1266
8.981.1	Class .....	1266
8.981.2	Slots .....	1266
8.982	Tootsville::Place .....	1267
8.982.1	Class .....	1267
8.982.2	Slots .....	1267
8.983	Tootsville::Place-Altitude .....	1268
8.983.1	Function .....	1268
8.983.2	File .....	1268
8.983.3	SetF Function .....	1268

8.983.4	File	1268
8.984	Tootsville::Place-Appearance	1269
8.984.1	Function	1269
8.984.2	File	1269
8.984.3	SetF Function	1269
8.984.4	File	1269
8.985	Tootsville::Place-Attributes	1270
8.985.1	Function	1270
8.985.2	File	1270
8.985.3	SetF Function	1270
8.985.4	File	1270
8.986	Tootsville::Place-Furniture	1271
8.986.1	Function	1271
8.986.2	File	1271
8.987	Tootsville::Place-Kind	1272
8.987.1	Function	1272
8.987.2	File	1272
8.987.3	SetF Function	1272
8.987.4	File	1272
8.988	Tootsville::Place-Latitude	1273
8.988.1	Function	1273
8.988.2	File	1273
8.988.3	SetF Function	1273
8.988.4	File	1273
8.989	Tootsville::Place-Longitude	1274
8.989.1	Function	1274
8.989.2	File	1274
8.989.3	SetF Function	1274
8.989.4	File	1274
8.990	Tootsville::Place-P	1275
8.990.1	Function	1275
8.990.2	File	1275
8.991	Tootsville::Place-Shape	1276
8.991.1	Function	1276
8.991.2	File	1276
8.991.3	SetF Function	1276
8.991.4	File	1276
8.992	Tootsville::Place-String	1277
8.992.1	Function	1277
8.992.2	File	1277
8.993	Tootsville::Place-String-Circle	1278
8.993.1	Function	1278
8.993.2	File	1278
8.994	Tootsville::Place-Uuid	1279
8.994.1	Function	1279
8.994.2	File	1279
8.994.3	SetF Function	1279
8.994.4	File	1279

8.995	Tootsville::Place-World	1280
8.995.1	Function	1280
8.995.2	File	1280
8.995.3	SetF Function	1280
8.995.4	File	1280
8.996	Tootsville::Places-At-Position	1281
8.996.1	Function	1281
8.996.2	File	1281
8.997	Tootsville::Play-With-Toot	1282
8.997.1	Function	1282
8.997.2	File	1282
8.998	Tootsville::Player-Adultp	1283
8.998.1	Function	1283
8.998.2	File	1283
8.999	Tootsville::Player-Alert	1284
8.999.1	Function	1284
8.999.2	File	1284
8.1000	Tootsville::Player-Childp	1285
8.1000.1	Function	1285
8.1000.2	File	1285
8.1001	Tootsville::Player-Toots	1286
8.1001.1	Function	1286
8.1001.2	File	1286
8.1002	Tootsville::Plist-To-English	1287
8.1002.1	Function	1287
8.1002.2	File	1287
8.1003	Tootsville::Plist-With-Index	1288
8.1003.1	Function	1288
8.1003.2	File	1288
8.1004	Tootsville::Point-Underwater-P	1289
8.1004.1	Function	1289
8.1004.2	File	1289
8.1005	Tootsville::Post-Sign-In	1290
8.1005.1	Function	1290
8.1005.2	File	1290
8.1006	Tootsville::Post/ Read-Version-Page	1291
8.1006.1	Function	1291
8.1006.2	File	1291
8.1007	Tootsville::Potential-Toot-Name-Character-P	1292
8.1007.1	Function	1292
8.1007.2	File	1292
8.1008	Tootsville::Potential-Toot-Name-P	1293
8.1008.1	Function	1293
8.1008.2	File	1293
8.1009	Tootsville::Power-On-Self-Test	1294
8.1009.1	Function	1294
8.1009.2	File	1294
8.1010	Tootsville::Powerset	1295

8.1010.1	Function	1295
8.1010.2	File	1295
8.1011	Tootsville::Pre-Login-Commands	1296
8.1011.1	Function	1296
8.1011.2	SetF Function	1296
8.1012	Tootsville::Precipitation	1297
8.1012.1	Function	1297
8.1012.2	File	1297
8.1013	Tootsville::Pretty-Print-Html-Error	1298
8.1013.1	Function	1298
8.1013.2	File	1298
8.1014	Tootsville::Print-Help	1299
8.1014.1	Function	1299
8.1014.2	File	1299
8.1015	Tootsville::Private-Admin-Message	1300
8.1015.1	Function	1300
8.1015.2	File	1300
8.1016	Tootsville::Prod	1301
8.1016.1	Variable	1301
8.1017	Tootsville::Props-Personality	1302
8.1017.1	Class	1302
8.1017.2	Slots	1302
8.1018	Tootsville::Pull-Records	1303
8.1018.1	Function	1303
8.1018.2	File	1303
8.1019	Tootsville::Pull-Records-Cache	1304
8.1019.1	Variable	1304
8.1020	Tootsville::Qa	1305
8.1020.1	Variable	1305
8.1021	Tootsville::Quaestor-Cancel-Event	1306
8.1021.1	Function	1306
8.1021.2	File	1306
8.1022	Tootsville::Quaestor-Complete-Event	1307
8.1022.1	Function	1307
8.1022.2	File	1307
8.1023	Tootsville::Quaestor-End-Fountain	1308
8.1023.1	Function	1308
8.1023.2	File	1308
8.1024	Tootsville::Quaestor-Event	1309
8.1024.1	Class	1309
8.1024.2	Slots	1309
8.1025	Tootsville::Quaestor-Event-Completedp	1310
8.1025.1	Function	1310
8.1025.2	File	1310
8.1025.3	SetF Function	1310
8.1025.4	File	1310
8.1026	Tootsville::Quaestor-Event-Ended-At	1311
8.1026.1	Function	1311

8.1026.2	File	1311
8.1026.3	SetF Function	1311
8.1026.4	File	1311
8.1027	Tootsville::Quaestor-Event-Fairy-Dust	1312
8.1027.1	Function	1312
8.1027.2	File	1312
8.1027.3	SetF Function	1312
8.1027.4	File	1312
8.1028	Tootsville::Quaestor-Event-Item	1313
8.1028.1	Function	1313
8.1028.2	File	1313
8.1028.3	SetF Function	1313
8.1028.4	File	1313
8.1029	Tootsville::Quaestor-Event-Medal	1314
8.1029.1	Function	1314
8.1029.2	File	1314
8.1029.3	SetF Function	1314
8.1029.4	File	1314
8.1030	Tootsville::Quaestor-Event-P	1315
8.1030.1	Function	1315
8.1030.2	File	1315
8.1031	Tootsville::Quaestor-Event-Peanuts	1316
8.1031.1	Function	1316
8.1031.2	File	1316
8.1031.3	SetF Function	1316
8.1031.4	File	1316
8.1032	Tootsville::Quaestor-Event-Score	1317
8.1032.1	Function	1317
8.1032.2	File	1317
8.1032.3	SetF Function	1317
8.1032.4	File	1317
8.1033	Tootsville::Quaestor-Event-Source	1318
8.1033.1	Function	1318
8.1033.2	File	1318
8.1033.3	SetF Function	1318
8.1033.4	File	1318
8.1034	Tootsville::Quaestor-Event-Started-At	1319
8.1034.1	Function	1319
8.1034.2	File	1319
8.1034.3	SetF Function	1319
8.1034.4	File	1319
8.1035	Tootsville::Quaestor-Event-Started-By	1320
8.1035.1	Function	1320
8.1035.2	File	1320
8.1035.3	SetF Function	1320
8.1035.4	File	1320
8.1036	Tootsville::Quaestor-Event-Uuid	1321
8.1036.1	Function	1321

8.1036.2	File	1321
8.1036.3	SetF Function	1321
8.1036.4	File	1321
8.1037	Tootsville::Quaestor-New-Toot	1322
8.1037.1	Function	1322
8.1037.2	File	1322
8.1038	Tootsville::Quaestor-Start-Event	1323
8.1038.1	Function	1323
8.1038.2	File	1323
8.1039	Tootsville::Quaestor-Start-General	1324
8.1039.1	Function	1324
8.1039.2	File	1324
8.1040	Tootsville::Query-Params	1325
8.1040.1	Function	1325
8.1040.2	File	1325
8.1041	Tootsville::Query-String->Plist	1326
8.1041.1	Function	1326
8.1041.2	File	1326
8.1042	Tootsville::Query-To-Memcache-Key	1327
8.1042.1	Function	1327
8.1042.2	File	1327
8.1043	Tootsville::Quiesce-Connected-Toots	1328
8.1043.1	Function	1328
8.1043.2	File	1328
8.1044	Tootsville::Rad-Personality	1329
8.1044.1	Class	1329
8.1044.2	Slots	1329
8.1045	Tootsville::Random-Key	1330
8.1045.1	Function	1330
8.1045.2	SetF Function	1330
8.1046	Tootsville::Random-Start-Wtl-For-Toot	1331
8.1046.1	Function	1331
8.1046.2	File	1331
8.1047	Tootsville::Raw-Post-String	1332
8.1047.1	Function	1332
8.1047.2	File	1332
8.1048	Tootsville::Read-Related-Journal	1333
8.1048.1	Function	1333
8.1048.2	File	1333
8.1049	Tootsville::Read-Staff-Journal	1334
8.1049.1	Function	1334
8.1049.2	File	1334
8.1050	Tootsville::Reap-Uninteresting-Child-Requests	1335
8.1050.1	Function	1335
8.1050.2	File	1335
8.1051	Tootsville::Reasonable-Name-Char-P	1336
8.1051.1	Function	1336
8.1051.2	File	1336

8.1052	Tootsville::Reasonable-Name-P .....	1337
8.1052.1	Function .....	1337
8.1052.2	File .....	1337
8.1053	Tootsville::Rebuild-Myself .....	1338
8.1053.1	Function .....	1338
8.1053.2	File .....	1338
8.1054	Tootsville::Redirect-To .....	1339
8.1054.1	Function .....	1339
8.1054.2	File .....	1339
8.1055	Tootsville::Redirect-To/ Html-Body .....	1340
8.1055.1	Function .....	1340
8.1055.2	File .....	1340
8.1056	Tootsville::Register-Metronome-Tasks .....	1341
8.1056.1	Function .....	1341
8.1056.2	File .....	1341
8.1057	Tootsville::Register-Signal-Handlers .....	1342
8.1057.1	Function .....	1342
8.1057.2	File .....	1342
8.1058	Tootsville::Relative-Facing .....	1343
8.1058.1	Function .....	1343
8.1058.2	File .....	1343
8.1059	Tootsville::Reload-Production .....	1344
8.1059.1	Function .....	1344
8.1059.2	File .....	1344
8.1060	Tootsville::Remap-Endpoints .....	1345
8.1060.1	Function .....	1345
8.1060.2	File .....	1345
8.1061	Tootsville::Remove-Furniture .....	1346
8.1061.1	Function .....	1346
8.1061.2	File .....	1346
8.1062	Tootsville::Remove-Repeats-For-Toot-Name .....	1347
8.1062.1	Function .....	1347
8.1062.2	File .....	1347
8.1063	Tootsville::Rename-Toot .....	1348
8.1063.1	Function .....	1348
8.1063.2	File .....	1348
8.1064	Tootsville::Render-Json .....	1349
8.1064.1	Function .....	1349
8.1064.2	File .....	1349
8.1065	Tootsville::Replace-TeXinfo-Tables .....	1350
8.1065.1	Function .....	1350
8.1065.2	File .....	1350
8.1066	Tootsville::Report-Slow-Query .....	1351
8.1066.1	Function .....	1351
8.1066.2	File .....	1351
8.1067	Tootsville::Request-Accept-Types .....	1352
8.1067.1	Function .....	1352
8.1067.2	File .....	1352



8.1068	Tootsville::Respond-To-Error	1353
8.1068.1	Function	1353
8.1068.2	File	1353
8.1069	Tootsville::Restore-Robot-Wtl	1354
8.1069.1	Function	1354
8.1069.2	File	1354
8.1070	Tootsville::Return-New-Apple	1355
8.1070.1	Function	1355
8.1070.2	File	1355
8.1071	Tootsville::Rgb-Bytes->Rgb	1356
8.1071.1	Function	1356
8.1071.2	File	1356
8.1072	Tootsville::Robot	1357
8.1072.1	Class	1357
8.1072.2	Slots	1357
8.1073	Tootsville::Robot-Broadcast	1358
8.1073.1	Function	1358
8.1073.2	File	1358
8.1074	Tootsville::Robot-Chaos	1359
8.1074.1	Class	1359
8.1074.2	Slots	1359
8.1075	Tootsville::Robot-Course	1360
8.1075.1	Function	1360
8.1075.2	SetF Function	1360
8.1076	Tootsville::Robot-Course-Wtl	1361
8.1076.1	Function	1361
8.1076.2	File	1361
8.1077	Tootsville::Robot-Cupid	1362
8.1077.1	Class	1362
8.1077.2	Slots	1362
8.1078	Tootsville::Robot-Doodle	1363
8.1078.1	Class	1363
8.1078.2	Slots	1363
8.1079	Tootsville::Robot-Dottie	1364
8.1079.1	Class	1364
8.1079.2	Slots	1364
8.1080	Tootsville::Robot-Flora	1365
8.1080.1	Class	1365
8.1080.2	Slots	1365
8.1081	Tootsville::Robot-Go-To	1366
8.1081.1	Function	1366
8.1082	Tootsville::Robot-Handle	1367
8.1082.1	Function	1367
8.1082.2	File	1367
8.1083	Tootsville::Robot-Harmony	1368
8.1083.1	Class	1368
8.1083.2	Slots	1368
8.1084	Tootsville::Robot-Has-Heard	1369

8.1084.1	Function	1369
8.1084.2	SetF Function	1369
8.1085	Tootsville::Robot-Heard	1370
8.1085.1	Function	1370
8.1085.2	File	1370
8.1086	Tootsville::Robot-Jack	1371
8.1086.1	Class	1371
8.1086.2	Slots	1371
8.1087	Tootsville::Robot-Lil-Mc	1372
8.1087.1	Class	1372
8.1087.2	Slots	1372
8.1088	Tootsville::Robot-Listen	1373
8.1088.1	Function	1373
8.1088.2	File	1373
8.1089	Tootsville::Robot-Match	1374
8.1089.1	Macro	1374
8.1089.2	File	1374
8.1090	Tootsville::Robot-Mayor-Louis	1375
8.1090.1	Class	1375
8.1090.2	Slots	1375
8.1091	Tootsville::Robot-Mode	1376
8.1091.1	Function	1376
8.1091.2	SetF Function	1376
8.1092	Tootsville::Robot-Moo	1377
8.1092.1	Class	1377
8.1092.2	Slots	1377
8.1093	Tootsville::Robot-Nevermind	1378
8.1093.1	Class	1378
8.1093.2	Slots	1378
8.1094	Tootsville::Robot-Picasso	1379
8.1094.1	Class	1379
8.1094.2	Slots	1379
8.1095	Tootsville::Robot-Position	1380
8.1095.1	Function	1380
8.1095.2	File	1380
8.1096	Tootsville::Robot-Props	1381
8.1096.1	Class	1381
8.1096.2	Slots	1381
8.1097	Tootsville::Robot-Rad	1382
8.1097.1	Class	1382
8.1097.2	Slots	1382
8.1098	Tootsville::Robot-Say	1383
8.1098.1	Function	1383
8.1099	Tootsville::Robot-Set-Mode	1384
8.1099.1	Macro	1384
8.1099.2	File	1384
8.1100	Tootsville::Robot-Shade	1385
8.1100.1	Class	1385

8.1100.2	Slots	1385
8.1101	Tootsville::Robot-Smudge	1386
8.1101.1	Class	1386
8.1101.2	Slots	1386
8.1102	Tootsville::Robot-Snowcone	1387
8.1102.1	Class	1387
8.1102.2	Slots	1387
8.1103	Tootsville::Robot-Sparkle	1388
8.1103.1	Class	1388
8.1103.2	Slots	1388
8.1104	Tootsville::Robot-Sploot	1389
8.1104.1	Class	1389
8.1104.2	Slots	1389
8.1105	Tootsville::Robot-Superstar	1390
8.1105.1	Class	1390
8.1105.2	Slots	1390
8.1106	Tootsville::Robot-Unicast	1391
8.1106.1	Function	1391
8.1106.2	File	1391
8.1107	Tootsville::Robot-Zap	1392
8.1107.1	Class	1392
8.1107.2	Slots	1392
8.1108	Tootsville::Robotp	1393
8.1108.1	Function	1393
8.1108.2	File	1393
8.1109	Tootsville::Romance-Ii-Copyright-Latest	1394
8.1109.1	Function	1394
8.1109.2	File	1394
8.1110	Tootsville::Romance-Ii-Program-Name	1395
8.1110.1	Function	1395
8.1110.2	File	1395
8.1111	Tootsville::Romance-Ii-Program-Name/ Version	1396
8.1111.1	Function	1396
8.1111.2	File	1396
8.1112	Tootsville::Romance-Ii-Program-Version	1397
8.1112.1	Function	1397
8.1112.2	File	1397
8.1113	Tootsville::Run-Async	1398
8.1113.1	Function	1398
8.1113.2	File	1398
8.1114	Tootsville::Run-Metronome-Tasks	1399
8.1114.1	Function	1399
8.1114.2	File	1399
8.1115	Tootsville::Save-Record	1400
8.1115.1	Function	1400
8.1115.2	File	1400
8.1116	Tootsville::Send-Parent-Child-Login-Email	1401
8.1116.1	Function	1401

8.1116.2	File	1401
8.1117	Tootsville::Send-Parent-Child-Login-Request	1402
8.1117.1	Function	1402
8.1117.2	File	1402
8.1118	Tootsville::Send-Reply-As-Bytes	1403
8.1118.1	Function	1403
8.1118.2	File	1403
8.1119	Tootsville::Send-Sms-Message	1404
8.1119.1	Function	1404
8.1119.2	File	1404
8.1120	Tootsville::Server-List	1405
8.1120.1	Function	1405
8.1120.2	File	1405
8.1121	Tootsville::Set-Http-Default-Headers	1406
8.1121.1	Function	1406
8.1121.2	File	1406
8.1122	Tootsville::Set-Up-For-Daemon/ Error-Output	1407
8.1122.1	Function	1407
8.1122.2	File	1407
8.1123	Tootsville::Set-Up-For-Daemon/ Log-Output	1408
8.1123.1	Function	1408
8.1123.2	File	1408
8.1124	Tootsville::Set-Up-For-Daemon/ Standard-Output	1409
8.1124.1	Function	1409
8.1124.2	File	1409
8.1125	Tootsville::Set-Up-For-Daemon/ Start-Logging	1410
8.1125.1	Function	1410
8.1125.2	File	1410
8.1126	Tootsville::Set-Up-For-Daemon/ Trace-Output	1411
8.1126.1	Function	1411
8.1126.2	File	1411
8.1127	Tootsville::Set-User-Var	1412
8.1127.1	Function	1412
8.1127.2	File	1412
8.1128	Tootsville::Set-User-Var-D	1413
8.1128.1	Function	1413
8.1128.2	File	1413
8.1129	Tootsville::Set-User-Var-Wtl	1414
8.1129.1	Function	1414
8.1129.2	File	1414
8.1130	Tootsville::Sha1-Hash	1415
8.1130.1	Function	1415
8.1130.2	File	1415
8.1131	Tootsville::Sha1-Hex	1416
8.1131.1	Function	1416
8.1131.2	File	1416
8.1132	Tootsville::Shaddow-Personality	1417
8.1132.1	Class	1417

8.1132.2	Slots	1417
8.1133	Tootsville::Shade-Personality	1418
8.1133.1	Class	1418
8.1133.2	Slots	1418
8.1134	Tootsville::Shift-Contour-Point	1419
8.1134.1	Function	1419
8.1134.2	File	1419
8.1135	Tootsville::Sinus	1420
8.1135.1	Function	1420
8.1135.2	File	1420
8.1136	Tootsville::Sky-Contents	1421
8.1136.1	Function	1421
8.1136.2	File	1421
8.1137	Tootsville::Sky-Room-Var	1422
8.1137.1	Function	1422
8.1137.2	Example structure	1422
8.1137.3	File	1422
8.1138	Tootsville::Slot-Values	1423
8.1138.1	Function	1423
8.1138.2	File	1423
8.1139	Tootsville::Smoothe-Contour-200×200	1424
8.1139.1	Function	1424
8.1139.2	File	1424
8.1140	Tootsville::Sms	1425
8.1140.1	Class	1425
8.1140.2	Slots	1425
8.1141	Tootsville::Sms-Destination	1426
8.1141.1	Function	1426
8.1141.2	File	1426
8.1141.3	SetF Function	1426
8.1141.4	File	1426
8.1142	Tootsville::Sms-Message	1427
8.1142.1	Function	1427
8.1142.2	File	1427
8.1142.3	SetF Function	1427
8.1142.4	File	1427
8.1143	Tootsville::Sms-Message-Index	1428
8.1143.1	Function	1428
8.1143.2	File	1428
8.1144	Tootsville::Sms-Mmsp	1429
8.1144.1	Function	1429
8.1144.2	File	1429
8.1144.3	SetF Function	1429
8.1144.4	File	1429
8.1145	Tootsville::Sms-P	1430
8.1145.1	Function	1430
8.1145.2	File	1430
8.1146	Tootsville::Sms-Sender	1431

8.1146.1	Function	1431
8.1146.2	File	1431
8.1146.3	SetF Function	1431
8.1146.4	File	1431
8.1147	Tootsville::Sms-Uuid	1432
8.1147.1	Function	1432
8.1147.2	File	1432
8.1147.3	SetF Function	1432
8.1147.4	File	1432
8.1148	Tootsville::Smudge-Personality	1433
8.1148.1	Class	1433
8.1148.2	Slots	1433
8.1149	Tootsville::Snowcone-Personality	1434
8.1149.1	Class	1434
8.1149.2	Slots	1434
8.1150	Tootsville::Sparkle-Personality	1435
8.1150.1	Class	1435
8.1150.2	Slots	1435
8.1151	Tootsville::Spawn-Terrain	1436
8.1151.1	Function	1436
8.1151.2	File	1436
8.1152	Tootsville::Split-Backtrace	1437
8.1152.1	Function	1437
8.1152.2	File	1437
8.1153	Tootsville::Split-Plist	1438
8.1153.1	Function	1438
8.1153.2	File	1438
8.1154	Tootsville::Sploot-Personality	1439
8.1154.1	Class	1439
8.1154.2	Slots	1439
8.1155	Tootsville::Square	1440
8.1155.1	Function	1440
8.1155.2	File	1440
8.1156	Tootsville::Ssl-Certificate	1441
8.1156.1	Function	1441
8.1156.2	File	1441
8.1157	Tootsville::Ssl-Private-Key	1442
8.1157.1	Function	1442
8.1157.2	File	1442
8.1158	Tootsville::Stamp-Toot-Passport	1443
8.1158.1	Function	1443
8.1158.2	File	1443
8.1159	Tootsville::Standard-Log-File	1444
8.1159.1	Function	1444
8.1159.2	File	1444
8.1160	Tootsville::Start	1445
8.1160.1	Function	1445
8.1160.2	File	1445

8.1161	Tootsville::Start-Game-Metronome .....	1446
8.1161.1	Function .....	1446
8.1161.2	File .....	1446
8.1162	Tootsville::Start-Hunchentoot .....	1447
8.1162.1	Function .....	1447
8.1162.2	File .....	1447
8.1163	Tootsville::Start-Production .....	1448
8.1163.1	Function .....	1448
8.1163.2	File .....	1448
8.1164	Tootsville::Start-Purchase-Event .....	1449
8.1164.1	Function .....	1449
8.1164.2	File .....	1449
8.1165	Tootsville::Start-Swank .....	1450
8.1165.1	Function .....	1450
8.1165.2	File .....	1450
8.1166	Tootsville::Start-Tcp-Listener .....	1451
8.1166.1	Function .....	1451
8.1166.2	File .....	1451
8.1167	Tootsville::Stop .....	1452
8.1167.1	Function .....	1452
8.1167.2	File .....	1452
8.1168	Tootsville::Stop-Game-Metronome .....	1453
8.1168.1	Function .....	1453
8.1168.2	File .....	1453
8.1169	Tootsville::Stop-Listening-For-Websockets .....	1454
8.1169.1	Function .....	1454
8.1169.2	File .....	1454
8.1170	Tootsville::Stop-Production .....	1455
8.1170.1	Function .....	1455
8.1170.2	File .....	1455
8.1171	Tootsville::Store-Info .....	1456
8.1171.1	Function .....	1456
8.1171.2	File .....	1456
8.1172	Tootsville::Store-Item .....	1457
8.1172.1	Class .....	1457
8.1172.2	Slots .....	1457
8.1173	Tootsville::Store-Item-Currency .....	1458
8.1173.1	Function .....	1458
8.1173.2	File .....	1458
8.1173.3	SetF Function .....	1458
8.1173.4	File .....	1458
8.1174	Tootsville::Store-Item-P .....	1459
8.1174.1	Function .....	1459
8.1174.2	File .....	1459
8.1175	Tootsville::Store-Item-Price .....	1460
8.1175.1	Function .....	1460
8.1175.2	File .....	1460
8.1175.3	SetF Function .....	1460

8.1175.4	File	1460
8.1176	Tootsville::Store-Item-Qty	1461
8.1176.1	Function	1461
8.1176.2	File	1461
8.1176.3	SetF Function	1461
8.1176.4	File	1461
8.1177	Tootsville::Store-Item-Template	1462
8.1177.1	Function	1462
8.1177.2	File	1462
8.1177.3	SetF Function	1462
8.1177.4	File	1462
8.1178	Tootsville::Store-Item-Uuid	1463
8.1178.1	Function	1463
8.1178.2	File	1463
8.1178.3	SetF Function	1463
8.1178.4	File	1463
8.1179	Tootsville::String-All-Alpha-Chars-P	1464
8.1179.1	Function	1464
8.1179.2	File	1464
8.1180	Tootsville::String-Length-2-P	1465
8.1180.1	Function	1465
8.1180.2	File	1465
8.1181	Tootsville::Strip-After-Sem	1466
8.1181.1	Function	1466
8.1181.2	File	1466
8.1182	Tootsville::Subheader-Field	1467
8.1182.1	Function	1467
8.1182.2	File	1467
8.1183	Tootsville::Sun-Position	1468
8.1183.1	Function	1468
8.1183.2	File	1468
8.1184	Tootsville::Superstar-Personality	1469
8.1184.1	Class	1469
8.1184.2	Slots	1469
8.1185	Tootsville::Swank-Connected-P	1470
8.1185.1	Function	1470
8.1185.2	File	1470
8.1186	Tootsville::Sync	1471
8.1186.1	Function	1471
8.1186.2	File	1471
8.1187	Tootsville::Take-Item	1472
8.1187.1	Function	1472
8.1187.2	File	1472
8.1188	Tootsville::Tcp-Bandwidth-Record	1473
8.1188.1	Function	1473
8.1188.2	File	1473
8.1189	Tootsville::Tcp-Broadcast	1474
8.1189.1	Function	1474



8.1189.2	File	1474
8.1190	Tootsville::Tcp-Client	1475
8.1190.1	Class	1475
8.1190.2	Slots	1475
8.1191	Tootsville::Tcp-Client-Buffer	1476
8.1191.1	Function	1476
8.1191.2	File	1476
8.1191.3	SetF Function	1476
8.1191.4	File	1476
8.1192	Tootsville::Tcp-Client-Expected-Length	1477
8.1192.1	Function	1477
8.1192.2	File	1477
8.1192.3	SetF Function	1477
8.1192.4	File	1477
8.1193	Tootsville::Tcp-Client-P	1478
8.1193.1	Function	1478
8.1193.2	File	1478
8.1194	Tootsville::Tcp-Client-Peer	1479
8.1194.1	Function	1479
8.1194.2	File	1479
8.1194.3	SetF Function	1479
8.1194.4	File	1479
8.1195	Tootsville::Tcp-Client-Socket	1480
8.1195.1	Function	1480
8.1195.2	File	1480
8.1195.3	SetF Function	1480
8.1195.4	File	1480
8.1196	Tootsville::Tcp-Format-Error	1481
8.1196.1	Function	1481
8.1196.2	File	1481
8.1197	Tootsville::Tcp-Handle-Peer-Request	1482
8.1197.1	Function	1482
8.1197.2	File	1482
8.1198	Tootsville::Tcp-Process-Packet	1483
8.1198.1	Function	1483
8.1198.2	File	1483
8.1199	Tootsville::Tcp-Reply	1484
8.1199.1	Function	1484
8.1199.2	File	1484
8.1200	Tootsville::Tcp-Socket-Input	1485
8.1200.1	Function	1485
8.1200.2	File	1485
8.1201	Tootsville::Tcp-Stream-Authenticate	1486
8.1201.1	Function	1486
8.1201.2	File	1486
8.1202	Tootsville::Tcp-Unicast	1487
8.1202.1	Function	1487
8.1202.2	File	1487

8.1203	Tootsville::Template->Openapi .....	1488
8.1203.1	Function .....	1488
8.1203.2	File .....	1488
8.1204	Tootsville::Template-Match .....	1489
8.1204.1	Function .....	1489
8.1204.2	File .....	1489
8.1205	Tootsville::Terrain .....	1490
8.1205.1	Function .....	1490
8.1205.2	File .....	1490
8.1206	Tootsville::Terrain-Db-Key .....	1491
8.1206.1	Function .....	1491
8.1206.2	File .....	1491
8.1207	Tootsville::Terrain-Exists-P .....	1492
8.1207.1	Function .....	1492
8.1207.2	File .....	1492
8.1208	Tootsville::Terrain-Height .....	1493
8.1208.1	Class .....	1493
8.1208.2	Slots .....	1493
8.1209	Tootsville::Terrain-Height-Latitude .....	1494
8.1209.1	Function .....	1494
8.1209.2	File .....	1494
8.1209.3	SetF Function .....	1494
8.1209.4	File .....	1494
8.1210	Tootsville::Terrain-Height-Longitude .....	1495
8.1210.1	Function .....	1495
8.1210.2	File .....	1495
8.1210.3	SetF Function .....	1495
8.1210.4	File .....	1495
8.1211	Tootsville::Terrain-Height-P .....	1496
8.1211.1	Function .....	1496
8.1211.2	File .....	1496
8.1212	Tootsville::Terrain-Height-Terrain .....	1497
8.1212.1	Function .....	1497
8.1212.2	File .....	1497
8.1212.3	SetF Function .....	1497
8.1212.4	File .....	1497
8.1213	Tootsville::Terrain-Height-World .....	1498
8.1213.1	Function .....	1498
8.1213.2	File .....	1498
8.1213.3	SetF Function .....	1498
8.1213.4	File .....	1498
8.1214	Tootsville::Terrain/ Add-Cactus .....	1499
8.1214.1	Function .....	1499
8.1214.2	File .....	1499
8.1215	Tootsville::Terrain/ Add-Flowers .....	1500
8.1215.1	Function .....	1500
8.1215.2	File .....	1500
8.1216	Tootsville::Terrain/ Add-Log .....	1501

8.1216.1	Function .....	1501
8.1216.2	File .....	1501
8.1217	Tootsville::Terrain/ Add-Mushrooms .....	1502
8.1217.1	Function .....	1502
8.1217.2	File .....	1502
8.1218	Tootsville::Terrain/ Add-Shaddow-Bush .....	1503
8.1218.1	Function .....	1503
8.1218.2	File .....	1503
8.1219	Tootsville::Terrain/ Add-Shaddow-Pit .....	1504
8.1219.1	Function .....	1504
8.1219.2	File .....	1504
8.1220	Tootsville::Terrain/ Add-Shaddow-Stalagmite .....	1505
8.1220.1	Function .....	1505
8.1220.2	File .....	1505
8.1221	Tootsville::Terrain/ Add-Small-Pond .....	1506
8.1221.1	Function .....	1506
8.1221.2	File .....	1506
8.1222	Tootsville::Terrain/ Add-Tree .....	1507
8.1222.1	Function .....	1507
8.1222.2	File .....	1507
8.1223	Tootsville::Terrain/ Connect-Streams .....	1508
8.1223.1	Function .....	1508
8.1223.2	File .....	1508
8.1224	Tootsville::Terrain/ Stream-Present-P .....	1509
8.1224.1	Function .....	1509
8.1224.2	File .....	1509
8.1225	Tootsville::Test .....	1510
8.1225.1	Variable .....	1510
8.1226	Tootsville::Texi-Ref .....	1511
8.1226.1	Function .....	1511
8.1226.2	File .....	1511
8.1227	Tootsville::Three-Chars-In-A-Row-P .....	1512
8.1227.1	Function .....	1512
8.1227.2	File .....	1512
8.1228	Tootsville::Tick-Weather-Day .....	1513
8.1228.1	Function .....	1513
8.1228.2	File .....	1513
8.1229	Tootsville::Tick-Weather-Minute .....	1514
8.1229.1	Function .....	1514
8.1229.2	File .....	1514
8.1230	Tootsville::Toot .....	1515
8.1230.1	Function .....	1515
8.1230.2	File .....	1515
8.1230.3	SetF Function .....	1515
8.1230.4	Class .....	1515
8.1230.5	Slots .....	1515
8.1231	Tootsville::Toot-Avatar .....	1516
8.1231.1	Function .....	1516

8.1231.2	File	1516
8.1231.3	SetF Function	1516
8.1231.4	File	1516
8.1232	Tootsville::Toot-Avatar-Scale-X	1517
8.1232.1	Function	1517
8.1232.2	File	1517
8.1232.3	SetF Function	1517
8.1232.4	File	1517
8.1233	Tootsville::Toot-Avatar-Scale-Y	1518
8.1233.1	Function	1518
8.1233.2	File	1518
8.1233.3	SetF Function	1518
8.1233.4	File	1518
8.1234	Tootsville::Toot-Avatar-Scale-Z	1519
8.1234.1	Function	1519
8.1234.2	File	1519
8.1234.3	SetF Function	1519
8.1234.4	File	1519
8.1235	Tootsville::Toot-Base-Color	1520
8.1235.1	Function	1520
8.1235.2	File	1520
8.1235.3	SetF Function	1520
8.1235.4	File	1520
8.1236	Tootsville::Toot-Base-Color-Name	1521
8.1236.1	Type	1521
8.1237	Tootsville::Toot-Base-Color-Name-P	1522
8.1237.1	Function	1522
8.1237.2	File	1522
8.1238	Tootsville::Toot-Buddy-List	1523
8.1238.1	Function	1523
8.1238.2	File	1523
8.1239	Tootsville::Toot-Can-Afford-P	1524
8.1239.1	Function	1524
8.1239.2	File	1524
8.1240	Tootsville::Toot-Chat-Background-Color	1525
8.1240.1	Function	1525
8.1240.2	File	1525
8.1241	Tootsville::Toot-Chat-Foreground-Color	1526
8.1241.1	Function	1526
8.1241.2	File	1526
8.1242	Tootsville::Toot-Child-Code	1527
8.1242.1	Function	1527
8.1242.2	File	1527
8.1242.3	SetF Function	1527
8.1242.4	File	1527
8.1243	Tootsville::Toot-Childp	1528
8.1243.1	Function	1528
8.1243.2	File	1528

8.1244	Tootsville::Toot-Clothes+Pattern.....	1529
8.1244.1	Function.....	1529
8.1244.2	File.....	1529
8.1245	Tootsville::Toot-Contacts.....	1530
8.1245.1	Function.....	1530
8.1245.2	File.....	1530
8.1246	Tootsville::Toot-Equipped-Item.....	1531
8.1246.1	Function.....	1531
8.1246.2	File.....	1531
8.1247	Tootsville::Toot-Fairy-Dust.....	1532
8.1247.1	Function.....	1532
8.1247.2	File.....	1532
8.1248	Tootsville::Toot-Has-Item-P.....	1533
8.1248.1	Function.....	1533
8.1248.2	File.....	1533
8.1249	Tootsville::Toot-Info.....	1534
8.1249.1	Function.....	1534
8.1249.2	Data Structure.....	1534
8.1249.3	Changes from 1.0 to 1.1.....	1535
8.1249.4	Changes from 1.1 to 1.2.....	1535
8.1249.5	Changes from 1.2 to 2.0.....	1536
8.1249.6	Deprecation.....	1536
8.1249.7	Obtaining Toot Information.....	1536
8.1249.8	File.....	1536
8.1250	Tootsville::Toot-Inventory.....	1537
8.1250.1	Function.....	1537
8.1250.2	File.....	1537
8.1251	Tootsville::Toot-Join-Message.....	1538
8.1251.1	Function.....	1538
8.1251.2	File.....	1538
8.1252	Tootsville::Toot-Last-Active.....	1539
8.1252.1	Function.....	1539
8.1252.2	File.....	1539
8.1252.3	SetF Function.....	1539
8.1252.4	File.....	1539
8.1253	Tootsville::Toot-List-Message.....	1540
8.1253.1	Function.....	1540
8.1253.2	Format.....	1540
8.1253.3	File.....	1540
8.1254	Tootsville::Toot-Name.....	1541
8.1254.1	Function.....	1541
8.1254.2	File.....	1541
8.1254.3	SetF Function.....	1541
8.1254.4	File.....	1541
8.1254.5	Type.....	1541
8.1255	Tootsville::Toot-Note.....	1542
8.1255.1	Function.....	1542
8.1255.2	File.....	1542

8.1255.3	SetF Function .....	1542
8.1255.4	File .....	1542
8.1256	Tootsville::Toot-P .....	1543
8.1256.1	Function .....	1543
8.1256.2	File .....	1543
8.1257	Tootsville::Toot-Pad-Color .....	1544
8.1257.1	Function .....	1544
8.1257.2	File .....	1544
8.1257.3	SetF Function .....	1544
8.1257.4	File .....	1544
8.1258	Tootsville::Toot-Pad-Color-Name .....	1545
8.1258.1	Type .....	1545
8.1259	Tootsville::Toot-Pad-Color-Name-P .....	1546
8.1259.1	Function .....	1546
8.1259.2	File .....	1546
8.1260	Tootsville::Toot-Passport-Stamped-P .....	1547
8.1260.1	Function .....	1547
8.1260.2	File .....	1547
8.1261	Tootsville::Toot-Passport-Stamps .....	1548
8.1261.1	Function .....	1548
8.1261.2	File .....	1548
8.1262	Tootsville::Toot-Pattern .....	1549
8.1262.1	Function .....	1549
8.1262.2	File .....	1549
8.1262.3	SetF Function .....	1549
8.1262.4	File .....	1549
8.1263	Tootsville::Toot-Pattern-Color .....	1550
8.1263.1	Function .....	1550
8.1263.2	File .....	1550
8.1263.3	SetF Function .....	1550
8.1263.4	File .....	1550
8.1264	Tootsville::Toot-Pattern-Color-Name .....	1551
8.1264.1	Type .....	1551
8.1265	Tootsville::Toot-Pattern-Color-Name-P .....	1552
8.1265.1	Function .....	1552
8.1265.2	File .....	1552
8.1266	Tootsville::Toot-Pattern-Name .....	1553
8.1266.1	Type .....	1553
8.1267	Tootsville::Toot-Pattern-Name-P .....	1554
8.1267.1	Function .....	1554
8.1267.2	File .....	1554
8.1268	Tootsville::Toot-Peanuts .....	1555
8.1268.1	Function .....	1555
8.1268.2	File .....	1555
8.1269	Tootsville::Toot-Personality .....	1556
8.1269.1	Class .....	1556
8.1269.2	Slots .....	1556
8.1270	Tootsville::Toot-Player .....	1557

8.1270.1	Function	1557
8.1270.2	File	1557
8.1270.3	SetF Function	1557
8.1270.4	File	1557
8.1271	Tootsville::Toot-Position	1558
8.1271.1	Function	1558
8.1271.2	File	1558
8.1271.3	SetF Function	1558
8.1272	Tootsville::Toot-Presentation-Name	1559
8.1272.1	Function	1559
8.1272.2	File	1559
8.1273	Tootsville::Toot-Private-Message	1560
8.1273.1	Function	1560
8.1273.2	File	1560
8.1274	Tootsville::Toot-Quiesced	1561
8.1274.1	Class	1561
8.1274.2	Slots	1561
8.1275	Tootsville::Toot-Quiesced-Altitude	1562
8.1275.1	Function	1562
8.1275.2	File	1562
8.1275.3	SetF Function	1562
8.1275.4	File	1562
8.1276	Tootsville::Toot-Quiesced-Attribs	1563
8.1276.1	Function	1563
8.1276.2	File	1563
8.1276.3	SetF Function	1563
8.1276.4	File	1563
8.1277	Tootsville::Toot-Quiesced-D3	1564
8.1277.1	Function	1564
8.1277.2	File	1564
8.1277.3	SetF Function	1564
8.1277.4	File	1564
8.1278	Tootsville::Toot-Quiesced-Data	1565
8.1278.1	Function	1565
8.1278.2	File	1565
8.1279	Tootsville::Toot-Quiesced-Emotion	1566
8.1279.1	Function	1566
8.1279.2	File	1566
8.1279.3	SetF Function	1566
8.1279.4	File	1566
8.1280	Tootsville::Toot-Quiesced-Latitude	1567
8.1280.1	Function	1567
8.1280.2	File	1567
8.1280.3	SetF Function	1567
8.1280.4	File	1567
8.1281	Tootsville::Toot-Quiesced-Longitude	1568
8.1281.1	Function	1568
8.1281.2	File	1568

8.1281.3	SetF Function .....	1568
8.1281.4	File .....	1568
8.1282	Tootsville::Toot-Quiesced-Observed .....	1569
8.1282.1	Function .....	1569
8.1282.2	File .....	1569
8.1282.3	SetF Function .....	1569
8.1282.4	File .....	1569
8.1283	Tootsville::Toot-Quiesced-P .....	1570
8.1283.1	Function .....	1570
8.1283.2	File .....	1570
8.1284	Tootsville::Toot-Quiesced-Peer-Address .....	1571
8.1284.1	Function .....	1571
8.1284.2	File .....	1571
8.1284.3	SetF Function .....	1571
8.1284.4	File .....	1571
8.1285	Tootsville::Toot-Quiesced-Toot .....	1572
8.1285.1	Function .....	1572
8.1285.2	File .....	1572
8.1285.3	SetF Function .....	1572
8.1285.4	File .....	1572
8.1286	Tootsville::Toot-Quiesced-World .....	1573
8.1286.1	Function .....	1573
8.1286.2	File .....	1573
8.1286.3	SetF Function .....	1573
8.1286.4	File .....	1573
8.1287	Tootsville::Toot-Quiesced-Wtl .....	1574
8.1287.1	Function .....	1574
8.1287.2	File .....	1574
8.1287.3	SetF Function .....	1574
8.1287.4	File .....	1574
8.1288	Tootsville::Toot-Sms-Messages .....	1575
8.1288.1	Function .....	1575
8.1288.2	File .....	1575
8.1289	Tootsville::Toot-Speak .....	1576
8.1289.1	Function .....	1576
8.1289.2	File .....	1576
8.1290	Tootsville::Toot-Uuid .....	1577
8.1290.1	Function .....	1577
8.1290.2	File .....	1577
8.1290.3	SetF Function .....	1577
8.1290.4	File .....	1577
8.1291	Tootsville::Tootsville-Rest-Acceptor .....	1578
8.1291.1	Class .....	1578
8.1291.2	Slots .....	1578
8.1292	Tootsville::Tootsville-Rest-Ssl-Acceptor .....	1579
8.1292.1	Class .....	1579
8.1292.2	Slots .....	1579
8.1293	Tootsville::Tootsville-V-Banner .....	1580



8.1293.1	Function	1580
8.1293.2	File	1580
8.1294	Tootsville::Trace-Log-File	1581
8.1294.1	Function	1581
8.1294.2	File	1581
8.1295	Tootsville::Trace-Output-Heartbeat	1582
8.1295.1	Function	1582
8.1295.2	File	1582
8.1296	Tootsville::Try-Reconnect-Toot-Name	1583
8.1296.1	Function	1583
8.1296.2	File	1583
8.1297	Tootsville::Two-Chars-In-A-Row-P	1584
8.1297.1	Function	1584
8.1297.2	File	1584
8.1298	Tootsville::Two-Letter-String	1585
8.1298.1	Type	1585
8.1299	Tootsville::Un-Banhammer-Ip-Address	1586
8.1299.1	Function	1586
8.1299.2	File	1586
8.1300	Tootsville::Unicast	1587
8.1300.1	Function	1587
8.1300.2	File	1587
8.1301	Tootsville::Unidentified-Player-Error	1588
8.1301.1	Class	1588
8.1301.2	Slots	1588
8.1302	Tootsville::Unimplemented	1589
8.1302.1	Class	1589
8.1302.2	Slots	1589
8.1303	Tootsville::Unimplemented-Feature	1590
8.1303.1	Function	1590
8.1303.2	SetF Function	1590
8.1304	Tootsville::Unprocessable	1591
8.1304.1	Class	1591
8.1304.2	Slots	1591
8.1305	Tootsville::Update-Gravatar	1592
8.1305.1	Function	1592
8.1305.2	File	1592
8.1306	Tootsville::Update-Nil	1593
8.1306.1	Class	1593
8.1306.2	Slots	1593
8.1307	Tootsville::Update-Toot-Last-Active	1594
8.1307.1	Function	1594
8.1307.2	File	1594
8.1308	Tootsville::Uri-To-Uuid	1595
8.1308.1	Function	1595
8.1308.2	File	1595
8.1309	Tootsville::Url-To-String	1596
8.1309.1	Function	1596

8.1309.2	File	1596
8.1310	Tootsville::User->Alist	1597
8.1310.1	Function	1597
8.1310.2	File	1597
8.1311	Tootsville::User-Account	1598
8.1311.1	Function	1598
8.1311.2	SetF Function	1598
8.1312	Tootsville::User-Display-Name	1599
8.1312.1	Function	1599
8.1312.2	File	1599
8.1313	Tootsville::User-Email	1600
8.1313.1	Function	1600
8.1313.2	File	1600
8.1314	Tootsville::User-Face	1601
8.1314.1	Function	1601
8.1314.2	File	1601
8.1315	Tootsville::User-Given-Name	1602
8.1315.1	Function	1602
8.1315.2	File	1602
8.1316	Tootsville::User-Id	1603
8.1316.1	Function	1603
8.1316.2	File	1603
8.1317	Tootsville::User-Online-P	1604
8.1317.1	Function	1604
8.1317.2	File	1604
8.1318	Tootsville::User-Stream	1605
8.1318.1	Function	1605
8.1318.2	File	1605
8.1319	Tootsville::User-Surname	1606
8.1319.1	Function	1606
8.1319.2	File	1606
8.1320	Tootsville::Uuid-String-P	1607
8.1320.1	Function	1607
8.1320.2	Example	1607
8.1320.3	File	1607
8.1321	Tootsville::Uuid-String-To-Base64	1608
8.1321.1	Function	1608
8.1321.2	File	1608
8.1322	Tootsville::Uuid-To-Base64	1609
8.1322.1	Function	1609
8.1322.2	File	1609
8.1323	Tootsville::Uuid-To-Uri	1610
8.1323.1	Function	1610
8.1323.2	File	1610
8.1324	Tootsville::Valid-Child-Code-P	1611
8.1324.1	Function	1611
8.1324.2	File	1611
8.1325	Tootsville::Value-To-Texi	1612

8.1325.1	Function	1612
8.1325.2	File	1612
8.1326	Tootsville::Vanish-Item	1613
8.1326.1	Function	1613
8.1326.2	File	1613
8.1327	Tootsville::Verbose-Log-File	1614
8.1327.1	Function	1614
8.1327.2	File	1614
8.1328	Tootsville::Version-Info-For	1615
8.1328.1	Function	1615
8.1328.2	File	1615
8.1329	Tootsville::Version-Info-List	1616
8.1329.1	Function	1616
8.1329.2	File	1616
8.1330	Tootsville::Version-Info-Report	1617
8.1330.1	Function	1617
8.1330.2	File	1617
8.1331	Tootsville::Version-Info-Report-String	1618
8.1331.1	Function	1618
8.1331.2	File	1618
8.1332	Tootsville::Wallet-Info	1619
8.1332.1	Function	1619
8.1332.2	Changes from 1.2 to 2.0	1619
8.1332.3	File	1619
8.1333	Tootsville::Wants-Json-P	1620
8.1333.1	Function	1620
8.1333.2	File	1620
8.1334	Tootsville::Wear-Slot	1621
8.1334.1	Class	1621
8.1334.2	Slots	1621
8.1335	Tootsville::Wear-Slot-Alternate	1622
8.1335.1	Function	1622
8.1335.2	File	1622
8.1335.3	SetF Function	1622
8.1335.4	File	1622
8.1336	Tootsville::Wear-Slot-Avatar-Point	1623
8.1336.1	Function	1623
8.1336.2	File	1623
8.1336.3	SetF Function	1623
8.1336.4	File	1623
8.1337	Tootsville::Wear-Slot-Id	1624
8.1337.1	Function	1624
8.1337.2	File	1624
8.1337.3	SetF Function	1624
8.1337.4	File	1624
8.1338	Tootsville::Wear-Slot-Info	1625
8.1338.1	Function	1625
8.1338.2	File	1625

8.1339	Tootsville::Wear-Slot-Name .....	1626
8.1339.1	Function .....	1626
8.1339.2	File .....	1626
8.1339.3	SetF Function .....	1626
8.1339.4	File .....	1626
8.1340	Tootsville::Wear-Slot-Obstruct-Max .....	1627
8.1340.1	Function .....	1627
8.1340.2	File .....	1627
8.1340.3	SetF Function .....	1627
8.1340.4	File .....	1627
8.1341	Tootsville::Wear-Slot-Obstruct-Min .....	1628
8.1341.1	Function .....	1628
8.1341.2	File .....	1628
8.1341.3	SetF Function .....	1628
8.1341.4	File .....	1628
8.1342	Tootsville::Wear-Slot-Obstruct-Point .....	1629
8.1342.1	Function .....	1629
8.1342.2	File .....	1629
8.1342.3	SetF Function .....	1629
8.1342.4	File .....	1629
8.1343	Tootsville::Wear-Slot-P .....	1630
8.1343.1	Function .....	1630
8.1343.2	File .....	1630
8.1344	Tootsville::Wear-Slot-Valence .....	1631
8.1344.1	Function .....	1631
8.1344.2	File .....	1631
8.1344.3	SetF Function .....	1631
8.1344.4	File .....	1631
8.1345	Tootsville::Websocket-Acceptor .....	1632
8.1345.1	Class .....	1632
8.1345.2	Slots .....	1632
8.1346	Tootsville::Websocket-Authenticate .....	1633
8.1346.1	Function .....	1633
8.1346.2	File .....	1633
8.1347	Tootsville::Websocket-Ssl-Acceptor .....	1634
8.1347.1	Class .....	1634
8.1347.2	Slots .....	1634
8.1348	Tootsville::Which-Toot-Is-Not-Yours .....	1635
8.1348.1	Function .....	1635
8.1348.2	SetF Function .....	1635
8.1349	Tootsville::Whitespace-Char-P .....	1636
8.1349.1	Function .....	1636
8.1349.2	File .....	1636
8.1350	Tootsville::Whitespacep .....	1637
8.1350.1	Function .....	1637
8.1350.2	File .....	1637
8.1351	Tootsville::Who-Is-Connected .....	1638
8.1351.1	Function .....	1638

8.1351.2	File	1638
8.1352	Tootsville::Wind-Vector	1639
8.1352.1	Class	1639
8.1352.2	Slots	1639
8.1353	Tootsville::Wind-Vector-P	1640
8.1353.1	Function	1640
8.1353.2	File	1640
8.1354	Tootsville::Wind-Vector-X-Magnitude	1641
8.1354.1	Function	1641
8.1354.2	File	1641
8.1354.3	SetF Function	1641
8.1354.4	File	1641
8.1355	Tootsville::Wind-Vector-Y-Magnitude	1642
8.1355.1	Function	1642
8.1355.2	File	1642
8.1355.3	SetF Function	1642
8.1355.4	File	1642
8.1356	Tootsville::Wind-X	1643
8.1356.1	Function	1643
8.1356.2	File	1643
8.1357	Tootsville::Wind-Y	1644
8.1357.1	Function	1644
8.1357.2	File	1644
8.1358	Tootsville::With-Cluster-Wide-Lock-Held	1645
8.1358.1	Macro	1645
8.1358.2	File	1645
8.1359	Tootsville::With-Continuable-Errors-Skipped	1646
8.1359.1	Macro	1646
8.1359.2	File	1646
8.1360	Tootsville::With-Dbi	1647
8.1360.1	Macro	1647
8.1360.2	File	1647
8.1361	Tootsville::With-Errors-As-Http	1648
8.1361.1	Macro	1648
8.1361.2	File	1648
8.1362	Tootsville::With-Http-Conditions	1649
8.1362.1	Macro	1649
8.1362.2	File	1649
8.1363	Tootsville::With-Http-Errors-As-Infinity-Errors	1650
8.1363.1	Macro	1650
8.1363.2	File	1650
8.1364	Tootsville::With-Local-Toot	1651
8.1364.1	Macro	1651
8.1364.2	File	1651
8.1365	Tootsville::With-Local-User	1652
8.1365.1	Macro	1652
8.1365.2	File	1652
8.1366	Tootsville::With-Maintenance-Times	1653

8.1366.1	Macro	1653
8.1366.2	File	1653
8.1367	Tootsville::With-Memcached-Query	1654
8.1367.1	Macro	1654
8.1367.2	File	1654
8.1368	Tootsville::With-Posted-Json	1655
8.1368.1	Macro	1655
8.1368.2	File	1655
8.1369	Tootsville::With-Score-In-Range	1656
8.1369.1	Macro	1656
8.1369.2	File	1656
8.1370	Tootsville::With-Standard-Streams-To-String	1657
8.1370.1	Macro	1657
8.1370.2	File	1657
8.1371	Tootsville::With-User	1658
8.1371.1	Macro	1658
8.1371.2	File	1658
8.1372	Tootsville::With-Websocket-Disconnections	1659
8.1372.1	Macro	1659
8.1372.2	File	1659
8.1373	Tootsville::Without-Medal	1660
8.1373.1	Macro	1660
8.1373.2	File	1660
8.1374	Tootsville::Without-Sem	1661
8.1374.1	Function	1661
8.1374.2	File	1661
8.1375	Tootsville::World	1662
8.1375.1	Function	1662
8.1375.2	File	1662
8.1375.3	Class	1662
8.1375.4	Slots	1662
8.1376	Tootsville::World-Mistp	1663
8.1376.1	Function	1663
8.1376.2	File	1663
8.1377	Tootsville::World-Moniker	1664
8.1377.1	Function	1664
8.1377.2	File	1664
8.1377.3	SetF Function	1664
8.1377.4	File	1664
8.1377.5	Type	1664
8.1378	Tootsville::World-Moniker-P	1665
8.1378.1	Function	1665
8.1378.2	File	1665
8.1379	Tootsville::World-Name	1666
8.1379.1	Function	1666
8.1379.2	File	1666
8.1379.3	SetF Function	1666
8.1379.4	File	1666

8.1380	Tootsville::World-P .....	1667
8.1380.1	Function .....	1667
8.1380.2	File .....	1667
8.1381	Tootsville::Write-Class-Docs .....	1668
8.1381.1	Function .....	1668
8.1381.2	File .....	1668
8.1382	Tootsville::Write-Docs .....	1669
8.1382.1	Function .....	1669
8.1382.2	File .....	1669
8.1383	Tootsville::Write-Docs-Header .....	1670
8.1383.1	Function .....	1670
8.1383.2	File .....	1670
8.1384	Tootsville::Write-Documentation .....	1671
8.1384.1	Function .....	1671
8.1384.2	File .....	1671
8.1385	Tootsville::Write-Function-Docs .....	1672
8.1385.1	Function .....	1672
8.1385.2	File .....	1672
8.1386	Tootsville::Write-Setf-Docs .....	1673
8.1386.1	Function .....	1673
8.1386.2	File .....	1673
8.1387	Tootsville::Write-Staff-Journal-Entry .....	1674
8.1387.1	Function .....	1674
8.1387.2	File .....	1674
8.1388	Tootsville::Ws-Approve-Toot .....	1675
8.1388.1	Function .....	1675
8.1388.2	File .....	1675
8.1389	Tootsville::Ws-Bandwidth-By-Source .....	1676
8.1389.1	Function .....	1676
8.1389.2	File .....	1676
8.1390	Tootsville::Ws-Bandwidth-Record .....	1677
8.1390.1	Function .....	1677
8.1390.2	File .....	1677
8.1391	Tootsville::Ws-Broadcast .....	1678
8.1391.1	Function .....	1678
8.1391.2	File .....	1678
8.1392	Tootsville::Ws-Client .....	1679
8.1392.1	Class .....	1679
8.1392.2	Slots .....	1679
8.1393	Tootsville::Ws-Deny-Toot .....	1680
8.1393.1	Function .....	1680
8.1393.2	File .....	1680
8.1394	Tootsville::Ws-Evacuate-All .....	1681
8.1394.1	Function .....	1681
8.1394.2	File .....	1681
8.1395	Tootsville::Ws-Kick .....	1682
8.1395.1	Function .....	1682
8.1395.2	File .....	1682

8.1396	Tootsville::Ws-Kick-Other-Streams-For-User	1683
8.1396.1	Function	1683
8.1396.2	File	1683
8.1397	Tootsville::Ws-Perform-Sign-In	1684
8.1397.1	Function	1684
8.1397.2	File	1684
8.1398	Tootsville::Ws-Reply	1685
8.1398.1	Function	1685
8.1398.2	File	1685
8.1399	Tootsville::Ws-Sign-In-User	1686
8.1399.1	Function	1686
8.1399.2	File	1686
8.1400	Tootsville::Ws-Stats	1687
8.1400.1	Function	1687
8.1400.2	File	1687
8.1401	Tootsville::Ws-Stats-Reset-All	1688
8.1401.1	Function	1688
8.1401.2	File	1688
8.1402	Tootsville::Ws-To-Infinity	1689
8.1402.1	Function	1689
8.1402.2	File	1689
8.1403	Tootsville::Ws-Unicast	1690
8.1403.1	Function	1690
8.1403.2	File	1690
8.1404	Tootsville::Ws-Without-Login	1691
8.1404.1	Function	1691
8.1404.2	File	1691
8.1405	Tootsville::Wtl-Course	1692
8.1405.1	Function	1692
8.1405.2	File	1692
8.1405.3	Class	1692
8.1405.4	Slots	1692
8.1406	Tootsville::Wtl-Course-Altitude	1693
8.1406.1	Function	1693
8.1406.2	File	1693
8.1406.3	SetF Function	1693
8.1406.4	File	1693
8.1407	Tootsville::Wtl-Course-End-Point	1694
8.1407.1	Function	1694
8.1407.2	File	1694
8.1407.3	SetF Function	1694
8.1407.4	File	1694
8.1408	Tootsville::Wtl-Course-End-Time	1695
8.1408.1	Function	1695
8.1408.2	File	1695
8.1408.3	SetF Function	1695
8.1408.4	File	1695
8.1409	Tootsville::Wtl-Course-Facing	1696



8.1409.1	Function	1696
8.1409.2	File	1696
8.1409.3	SetF Function	1696
8.1409.4	File	1696
8.1410	Tootsville::Wtl-Course-Latitude	1697
8.1410.1	Function	1697
8.1410.2	File	1697
8.1410.3	SetF Function	1697
8.1410.4	File	1697
8.1411	Tootsville::Wtl-Course-Longitude	1698
8.1411.1	Function	1698
8.1411.2	File	1698
8.1411.3	SetF Function	1698
8.1411.4	File	1698
8.1412	Tootsville::Wtl-Course-P	1699
8.1412.1	Function	1699
8.1412.2	File	1699
8.1413	Tootsville::Wtl-Course-Speed	1700
8.1413.1	Function	1700
8.1413.2	File	1700
8.1413.3	SetF Function	1700
8.1413.4	File	1700
8.1414	Tootsville::Wtl-Course-Start-Point	1701
8.1414.1	Function	1701
8.1414.2	File	1701
8.1414.3	SetF Function	1701
8.1414.4	File	1701
8.1415	Tootsville::Wtl-Course-Start-Time	1702
8.1415.1	Function	1702
8.1415.2	File	1702
8.1415.3	SetF Function	1702
8.1415.4	File	1702
8.1416	Tootsville::Wtl-Course-World	1703
8.1416.1	Function	1703
8.1416.2	File	1703
8.1416.3	SetF Function	1703
8.1416.4	File	1703
8.1417	Tootsville::Www-Uri	1704
8.1417.1	Type	1704
8.1418	Tootsville::Www-Uri-Like-P	1705
8.1418.1	Function	1705
8.1418.2	File	1705
8.1419	Tootsville::Yesterday	1706
8.1419.1	Function	1706
8.1419.2	File	1706
8.1420	Tootsville::Yield-Mariadb-Lock	1707
8.1420.1	Function	1707
8.1420.2	File	1707

8.1421	Tootsville::Zap-Personality .....	1708
8.1421.1	Class .....	1708
8.1421.2	Slots .....	1708
8.1422	Tootsville::⊕Post-Accept-Type-Does-Not-Match-/ *-When-Not-Allow-Wildcards-P .....	1709
8.1422.1	Function .....	1709
8.1422.2	File .....	1709
8.1423	Tootsville::⊕Post-Accept-Type-Matches-*/ * .....	1710
8.1423.1	Function .....	1710
8.1423.2	File .....	1710
8.1424	Tootsville::⊕Post-Accept-Type-Matches-/* .....	1711
8.1424.1	Function .....	1711
8.1424.2	File .....	1711
8.1425	Tootsville::⊕Post-Accept-Type-Matches-/*-With-Charset=UTF-8 ..	1712
8.1425.1	Function .....	1712
8.1425.2	File .....	1712
8.1426	Tootsville::⊕Post-Accept-Type-Matches-Identically .....	1713
8.1426.1	Function .....	1713
8.1426.2	File .....	1713
8.1427	Tootsville::⊕Post-Accept-Type-Matches-With-Charset=UTF-8 ..	1714
8.1427.1	Function .....	1714
8.1427.2	File .....	1714
8.1428	Tootsville::⊕Post-Acceptor-Template-Matches-Constants ..	1715
8.1428.1	Function .....	1715
8.1428.2	File .....	1715
8.1429	Tootsville::⊕Post-Acceptor-Template-Unifies-Variables .....	1716
8.1429.1	Function .....	1716
8.1429.2	File .....	1716
8.1430	Tootsville::⊕Post-Certificate-Extraction .....	1717
8.1430.1	Function .....	1717
8.1430.2	File .....	1717
8.1431	Tootsville::⊕Post-Check-Map-Heights .....	1718
8.1431.1	Function .....	1718
8.1431.2	File .....	1718
8.1432	Tootsville::⊕Post-Check-Map-Widths .....	1719
8.1432.1	Function .....	1719
8.1432.2	File .....	1719
8.1433	Tootsville::⊕Post-Ensure-Package-Imports-From-Oliphant-Are-Available ..	1720
8.1433.1	Function .....	1720
8.1433.2	File .....	1720
8.1434	Tootsville::⊕Post-Extract-Plist-Path-1 .....	1721
8.1434.1	Function .....	1721
8.1434.2	File .....	1721
8.1435	Tootsville::⊕Post-Extract-Plist-Path-2 .....	1722
8.1435.1	Function .....	1722
8.1435.2	File .....	1722

8.1436	Tootsville::⊕Post-Extract-Plist-Path-3.....	1723
8.1436.1	Function.....	1723
8.1436.2	File.....	1723
8.1437	Tootsville::⊕Post-Extract-Plist-Path-4.....	1724
8.1437.1	Function.....	1724
8.1437.2	File.....	1724
8.1438	Tootsville::⊕Post-Good-Uri-Amazon-S3.....	1725
8.1438.1	Function.....	1725
8.1438.2	File.....	1725
8.1439	Tootsville::⊕Post-Good-Uri-Tootsville.Org.....	1726
8.1439.1	Function.....	1726
8.1439.2	File.....	1726
8.1440	Tootsville::⊕Post-Good-Uri-With-Query-String.....	1727
8.1440.1	Function.....	1727
8.1440.2	File.....	1727
8.1441	Tootsville::⊕Post-Group-Plists.....	1728
8.1441.1	Function.....	1728
8.1441.2	File.....	1728
8.1442	Tootsville::⊕Post-Host-Name-Like-S3.Amazonaws.Com...	1729
8.1442.1	Function.....	1729
8.1442.2	File.....	1729
8.1443	Tootsville::⊕Post-Host-Name-Like-Star-Hope.Org.....	1730
8.1443.1	Function.....	1730
8.1443.2	File.....	1730
8.1444	Tootsville::⊕Post-Host-Name-Like-Tootsville.Org.....	1731
8.1444.1	Function.....	1731
8.1444.2	File.....	1731
8.1445	Tootsville::⊕Post-Host-Name-Like-Www.Gov.Uk.....	1732
8.1445.1	Function.....	1732
8.1445.2	File.....	1732
8.1446	Tootsville::⊕Post-Host-Name-Like-Www.Tootsville.Org...	1733
8.1446.1	Function.....	1733
8.1446.2	File.....	1733
8.1447	Tootsville::⊕Post-Memcached-Quick-Test.....	1734
8.1447.1	Function.....	1734
8.1447.2	File.....	1734
8.1448	Tootsville::⊕Post-Memcached-Random-Number-Test.....	1735
8.1448.1	Function.....	1735
8.1448.2	File.....	1735
8.1449	Tootsville::⊕Post-Normalize-Url-Collapse- / -To- /.....	1736
8.1449.1	Function.....	1736
8.1449.2	File.....	1736
8.1450	Tootsville::⊕Post-Normalize-Url-Collapse- / / -To- /.....	1737
8.1450.1	Function.....	1737
8.1450.2	File.....	1737
8.1451	Tootsville::⊕Post-Normalize-Url-Handle-./ -Chains.....	1738
8.1451.1	Function.....	1738
8.1451.2	File.....	1738

8.1452	Tootsville::⊕Post-Normalize-Url-Hostname-Downcased . . .	1739
8.1452.1	Function . . . . .	1739
8.1452.2	File . . . . .	1739
8.1453	Tootsville::⊕Post-Normalize-Url-Include-Unusual-Http-Port . .	1740
8.1453.1	Function . . . . .	1740
8.1453.2	File . . . . .	1740
8.1454	Tootsville::⊕Post-Normalize-Url-Include-Unusual-Https-Port . .	1741
8.1454.1	Function . . . . .	1741
8.1454.2	File . . . . .	1741
8.1455	Tootsville::⊕Post-Normalize-Url-Leave-%Xx-Encoded-Bytes . .	1742
8.1455.1	Function . . . . .	1742
8.1455.2	File . . . . .	1742
8.1456	Tootsville::⊕Post-Normalize-Url-Omit-Default-Https-Port . .	1743
8.1456.1	Function . . . . .	1743
8.1456.2	File . . . . .	1743
8.1457	Tootsville::⊕Post-Normalize-Url-Omit-Default-Http-Port . .	1744
8.1457.1	Function . . . . .	1744
8.1457.2	File . . . . .	1744
8.1458	Tootsville::⊕Post-Normalize-Url-Protocol-Downcased . . . . .	1745
8.1458.1	Function . . . . .	1745
8.1458.2	File . . . . .	1745
8.1459	Tootsville::⊕Post-Normalize-Url-Treat-./ -As-Up . . . . .	1746
8.1459.1	Function . . . . .	1746
8.1459.2	File . . . . .	1746
8.1460	Tootsville::⊕Post-Normalize-Url-Un%Xx-Escape-Basic-Ascii . .	1747
8.1460.1	Function . . . . .	1747
8.1460.2	File . . . . .	1747
8.1461	Tootsville::⊕Post-Normalize-Url-Use-%20-Not-+-For-Space . .	1748
8.1461.1	Function . . . . .	1748
8.1461.2	File . . . . .	1748
8.1462	Tootsville::⊕Post-Not-Host-Name-Like-Foo.Com . . . . .	1749
8.1462.1	Function . . . . .	1749
8.1462.2	File . . . . .	1749
8.1463	Tootsville::⊕Post-Not-Host-Name-Like-10.0.0.10 . . . . .	1750
8.1463.1	Function . . . . .	1750
8.1463.2	File . . . . .	1750
8.1464	Tootsville::⊕Post-Not-Host-Name-Like-9foo.Com . . . . .	1751
8.1464.1	Function . . . . .	1751
8.1464.2	File . . . . .	1751
8.1465	Tootsville::⊕Post-Not-Host-Name-Like-Bar.-Foo.Com . . . . .	1752
8.1465.1	Function . . . . .	1752
8.1465.2	File . . . . .	1752
8.1466	Tootsville::⊕Post-Not-Host-Name-Like-Bar.9foo.Com . . . . .	1753
8.1466.1	Function . . . . .	1753
8.1466.2	File . . . . .	1753
8.1467	Tootsville::⊕Post-Not-Host-Name-Like-Foo . . . . .	1754
8.1467.1	Function . . . . .	1754
8.1467.2	File . . . . .	1754

8.1468	Tootsville::⊕Post-Not-Host-Name-Like-Foo-Foo.Com . . . . .	1755
8.1468.1	Function . . . . .	1755
8.1468.2	File . . . . .	1755
8.1469	Tootsville::⊕Post-Not-Host-Name-Like-Foo-Com . . . . .	1756
8.1469.1	Function . . . . .	1756
8.1469.2	File . . . . .	1756
8.1470	Tootsville::⊕Post-Not-Host-Name-Like-Foo.12 . . . . .	1757
8.1470.1	Function . . . . .	1757
8.1470.2	File . . . . .	1757
8.1471	Tootsville::⊕Post-Not-Host-Name-Like-Foo.X . . . . .	1758
8.1471.1	Function . . . . .	1758
8.1471.2	File . . . . .	1758
8.1472	Tootsville::⊕Post-Not-Host-Name-Like-Ko . . . . .	1759
8.1472.1	Function . . . . .	1759
8.1472.2	File . . . . .	1759
8.1473	Tootsville::⊕Post-Subheader-Field-Parses . . . . .	1760
8.1473.1	Function . . . . .	1760
8.1473.2	File . . . . .	1760
8.1474	Tootsville::⊕Post-Unit-Test-Flatten-Plist-Tree . . . . .	1761
8.1474.1	Function . . . . .	1761
8.1474.2	File . . . . .	1761

## 9 Package Twilio . . . . . 1763

9.1	Twilio::As-Response . . . . .	1764
9.1.1	Macro . . . . .	1764
9.1.2	File . . . . .	1764
9.2	Twilio::Dial . . . . .	1765
9.2.1	Function . . . . .	1765
9.2.2	File . . . . .	1765
9.3	Twilio::Enqueue . . . . .	1766
9.3.1	Function . . . . .	1766
9.3.2	File . . . . .	1766
9.4	Twilio::Format-Language . . . . .	1767
9.4.1	Function . . . . .	1767
9.4.2	File . . . . .	1767
9.5	Twilio::Hangup . . . . .	1768
9.5.1	Function . . . . .	1768
9.5.2	File . . . . .	1768
9.6	Twilio::Leave . . . . .	1769
9.6.1	Function . . . . .	1769
9.6.2	File . . . . .	1769
9.7	Twilio::Message . . . . .	1770
9.7.1	Function . . . . .	1770
9.7.2	File . . . . .	1770
9.8	Twilio::Pause . . . . .	1771
9.8.1	Function . . . . .	1771
9.8.2	File . . . . .	1771
9.9	Twilio::Play . . . . .	1772

9.9.1	Function	1772
9.9.2	File	1772
9.10	Twilio::Play-Digits	1773
9.10.1	Function	1773
9.10.2	File	1773
9.11	Twilio::Record	1774
9.11.1	Function	1774
9.11.2	File	1774
9.12	Twilio::Redirect	1775
9.12.1	Function	1775
9.12.2	File	1775
9.13	Twilio::Reject	1776
9.13.1	Function	1776
9.13.2	File	1776
9.14	Twilio::Say	1777
9.14.1	Function	1777
9.14.2	File	1777
9.15	Twilio::With-Gather	1778
9.15.1	Macro	1778
9.15.2	File	1778
9.16	Twilio::With-Twilio-Params	1779
9.16.1	Macro	1779
9.16.2	File	1779
<b>10</b>	<b>Javascript</b>	<b>1781</b>
10.1	Tootsville.AvatarBuilder.addNameTag	1782
10.1.1	Function	1782
10.2	Tootsville.AvatarBuilder.build	1783
10.2.1	Function	1783
10.3	Tootsville.AvatarBuilder.colorize	1784
10.3.1	Function	1784
10.4	Tootsville.AvatarBuilder.enableShadows	1785
10.4.1	Function	1785
10.5	Tootsville.AvatarBuilder.getPathForPattern	1786
10.5.1	Function	1786
10.6	Tootsville.AvatarBuilder.loadAvatarBase	1787
10.6.1	Function	1787
10.7	Tootsville.AvatarBuilder.patterns	1788
10.7.1	Variable	1788
10.8	Tootsville.AvatarBuilder.postBuild	1789
10.8.1	Function	1789
10.9	Tootsville.AvatarBuilder.rainbowColor	1790
10.9.1	Function	1790
10.10	Tootsville.AvatarBuilder.rememberAvatar	1791
10.10.1	Function	1791
10.11	Tootsville.AvatarBuilder.update	1792
10.11.1	Function	1792
10.12	Tootsville.AvatarViewer.createCamera	1793

10.12.1	Function	1793
10.13	Tootsville.AvatarViewer.createLight	1794
10.13.1	Function	1794
10.14	Tootsville.AvatarViewer.createScene	1795
10.14.1	Function	1795
10.15	Tootsville.AvatarViewer.createViewerInCanvas	1796
10.15.1	Function	1796
10.16	Tootsville.AvatarViewer.createViewerReally	1797
10.16.1	Function	1797
10.17	Tootsville.AvatarViewer.getAvatar	1798
10.17.1	Function	1798
10.18	Tootsville.FurnitureBuilder.build	1799
10.18.1	Function	1799
10.19	Tootsville.FurnitureBuilder.build2	1800
10.19.1	Function	1800
10.20	Tootsville.FurnitureBuilder.colorize	1801
10.20.1	Function	1801
10.21	Tootsville.FurnitureBuilder.enableShadows	1802
10.21.1	Function	1802
10.22	Tootsville.FurnitureBuilder.loadItemTemplate	1803
10.22.1	Function	1803
10.23	Tootsville.FurnitureBuilder.rememberItem	1804
10.23.1	Function	1804
10.24	Tootsville.FurnitureBuilder.update	1805
10.24.1	Function	1805
10.25	Tootsville.Game.BallSystem.fastForward	1806
10.25.1	Function	1806
10.26	Tootsville.Game.BallSystem.register	1807
10.26.1	Function	1807
10.27	Tootsville.Game.BallSystem.remove	1808
10.27.1	Function	1808
10.28	Tootsville.Game.BallSystem.updateBalls	1809
10.28.1	Function	1809
10.29	Tootsville.Game.Commands.addFurniture	1810
10.29.1	Function	1810
10.30	Tootsville.Game.Commands.addToList	1811
10.30.1	Function	1811
10.31	Tootsville.Game.Commands.click	1812
10.31.1	Function	1812
10.32	Tootsville.Game.Commands.createUserHouse	1813
10.32.1	Function	1813
10.33	Tootsville.Game.Commands.doff	1814
10.33.1	Function	1814
10.34	Tootsville.Game.Commands.don	1815
10.34.1	Function	1815
10.35	Tootsville.Game.Commands.echo	1816
10.35.1	Function	1816
10.36	Tootsville.Game.Commands.endEvent	1817

10.36.1	Function	1817
10.37	Tootsville.Game.Commands.endevent	1818
10.37.1	Function	1818
10.38	Tootsville.Game.Commands.finger	1819
10.38.1	Function	1819
10.39	Tootsville.Game.Commands.gameAction	1820
10.39.1	Function	1820
10.40	Tootsville.Game.Commands.getApple	1821
10.40.1	Function	1821
10.41	Tootsville.Game.Commands.getAvatars	1822
10.41.1	Function	1822
10.42	Tootsville.Game.Commands.getColorPalettes	1823
10.42.1	Function	1823
10.42.2	410 Gone	1823
10.42.3	Note	1823
10.43	Tootsville.Game.Commands.getInventory	1824
10.43.1	Function	1824
10.44	Tootsville.Game.Commands.getInventoryByType	1825
10.44.1	Function	1825
10.45	Tootsville.Game.Commands.getOnlineUsers	1826
10.45.1	Function	1826
10.46	Tootsville.Game.Commands.getRoomList	1827
10.46.1	Function	1827
10.47	Tootsville.Game.Commands.getServerTime	1828
10.47.1	Function	1828
10.48	Tootsville.Game.Commands.getSessionApple	1829
10.48.1	Function	1829
10.49	Tootsville.Game.Commands.getStoreItemInfo	1830
10.49.1	Function	1830
10.50	Tootsville.Game.Commands.getUserLists	1831
10.50.1	Function	1831
10.51	Tootsville.Game.Commands.getWallet	1832
10.51.1	Function	1832
10.52	Tootsville.Game.Commands.getZoneList	1833
10.52.1	Function	1833
10.53	Tootsville.Game.Commands.give	1834
10.53.1	Function	1834
10.54	Tootsville.Game.Commands.go	1835
10.54.1	Function	1835
10.55	Tootsville.Game.Commands.initUserRoom	1836
10.55.1	Function	1836
10.56	Tootsville.Game.Commands.join	1837
10.56.1	Function	1837
10.57	Tootsville.Game.Commands.login	1838
10.57.1	Function	1838
10.58	Tootsville.Game.Commands.logout	1839
10.58.1	Function	1839
10.59	Tootsville.Game.Commands.mailCustomerService	1840



10.59.1	Function	1840
10.60	Tootsville.Game.Commands.peekAtInventory	1841
10.60.1	Function	1841
10.61	Tootsville.Game.Commands.ping	1842
10.61.1	Function	1842
10.62	Tootsville.Game.Commands.promptReply	1843
10.62.1	Function	1843
10.63	Tootsville.Game.Commands.removeFromList	1844
10.63.1	Function	1844
10.64	Tootsville.Game.Commands.reportBug	1845
10.64.1	Function	1845
10.65	Tootsville.Game.Commands.reportUser	1846
10.65.1	Function	1846
10.66	Tootsville.Game.Commands.requestBuddy	1847
10.66.1	Function	1847
10.67	Tootsville.Game.Commands.sendOutOfBandMessage	1848
10.67.1	Function	1848
10.68	Tootsville.Game.Commands.serverTime	1849
10.68.1	Function	1849
10.69	Tootsville.Game.Commands.setAvatarColor	1850
10.69.1	Function	1850
10.70	Tootsville.Game.Commands.setFurniture	1851
10.70.1	Function	1851
10.71	Tootsville.Game.Commands.spawnZone	1852
10.71.1	Function	1852
10.72	Tootsville.Game.Commands.speak	1853
10.72.1	Function	1853
10.73	Tootsville.Game.Commands.startEvent	1854
10.73.1	Function	1854
10.74	Tootsville.Game.Commands.useEquipment	1855
10.74.1	Function	1855
10.75	Tootsville.Game.Commands.walk	1856
10.75.1	Function	1856
10.75.2	Added in Romance 1.2	1856
10.75.3	Gossipnet only	1856
10.76	Tootsville.Game.Gatekeeper.admin	1857
10.76.1	Function	1857
10.77	Tootsville.Game.Gatekeeper.avatars	1858
10.77.1	Function	1858
10.78	Tootsville.Game.Gatekeeper.ayt	1859
10.78.1	Function	1859
10.79	Tootsville.Game.Gatekeeper.badgeUpdate	1860
10.79.1	Function	1860
10.80	Tootsville.Game.Gatekeeper.beam	1861
10.80.1	Function	1861
10.81	Tootsville.Game.Gatekeeper.bots	1862
10.81.1	Function	1862
10.82	Tootsville.Game.Gatekeeper.buddyList	1863

10.82.1	Function .....	1863
10.83	Tootsville.Game.Gatekeeper.buddyRequest .....	1864
10.83.1	Function .....	1864
10.83.2	Example .....	1864
10.84	Tootsville.Game.Gatekeeper.burgeon .....	1865
10.84.1	Function .....	1865
10.85	Tootsville.Game.Gatekeeper.bye .....	1866
10.85.1	Function .....	1866
10.86	Tootsville.Game.Gatekeeper.c .....	1867
10.86.1	Function .....	1867
10.87	Tootsville.Game.Gatekeeper.earning .....	1868
10.87.1	Function .....	1868
10.88	Tootsville.Game.Gatekeeper.endEvent .....	1869
10.88.1	Function .....	1869
10.89	Tootsville.Game.Gatekeeper.forceMove .....	1870
10.89.1	Function .....	1870
10.90	Tootsville.Game.Gatekeeper.gameAction .....	1871
10.90.1	Function .....	1871
10.91	Tootsville.Game.Gatekeeper.getApple .....	1872
10.91.1	Function .....	1872
10.92	Tootsville.Game.Gatekeeper.getAvailableHouses .....	1873
10.92.1	Function .....	1873
10.93	Tootsville.Game.Gatekeeper.getAwardRankings .....	1874
10.93.1	Function .....	1874
10.94	Tootsville.Game.Gatekeeper.getColorPalettes .....	1875
10.94.1	Function .....	1875
10.95	Tootsville.Game.Gatekeeper.getMailInBox .....	1876
10.95.1	Function .....	1876
10.96	Tootsville.Game.Gatekeeper.getMailMessage .....	1877
10.96.1	Function .....	1877
10.97	Tootsville.Game.Gatekeeper.getStoreItems .....	1878
10.97.1	Function .....	1878
10.98	Tootsville.Game.Gatekeeper.getUserLists .....	1879
10.98.1	Function .....	1879
10.99	Tootsville.Game.Gatekeeper.goToWeb .....	1880
10.99.1	Function .....	1880
10.100	Tootsville.Game.Gatekeeper.initUserRoom .....	1881
10.100.1	Function .....	1881
10.101	Tootsville.Game.Gatekeeper.inventory .....	1882
10.101.1	Function .....	1882
10.102	Tootsville.Game.Gatekeeper.joinOK .....	1883
10.102.1	Function .....	1883
10.103	Tootsville.Game.Gatekeeper.kick .....	1884
10.103.1	Function .....	1884
10.104	Tootsville.Game.Gatekeeper.logOK .....	1885
10.104.1	Function .....	1885
10.105	Tootsville.Game.Gatekeeper.login .....	1886
10.105.1	Function .....	1886

10.106	Tootsville.Game.Gatekeeper.migrate	1887
10.106.1	Function	1887
10.107	Tootsville.Game.Gatekeeper.newScript	1888
10.107.1	Function	1888
10.108	Tootsville.Game.Gatekeeper.outOfBand	1889
10.108.1	Function	1889
10.108.2	Invitation	1889
10.108.3	Response	1889
10.108.4	To Room	1889
10.109	Tootsville.Game.Gatekeeper.parentApproval	1890
10.109.1	Function	1890
10.110	Tootsville.Game.Gatekeeper.passport	1891
10.110.1	Function	1891
10.111	Tootsville.Game.Gatekeeper.playWith	1892
10.111.1	Function	1892
10.112	Tootsville.Game.Gatekeeper.postman	1893
10.112.1	Function	1893
10.113	Tootsville.Game.Gatekeeper.prompt	1894
10.113.1	Function	1894
10.114	Tootsville.Game.Gatekeeper.pub	1895
10.114.1	Function	1895
10.114.2	Example	1895
10.115	Tootsville.Game.Gatekeeper.purchase	1896
10.115.1	Function	1896
10.116	Tootsville.Game.Gatekeeper.quiesce	1897
10.116.1	Function	1897
10.117	Tootsville.Game.Gatekeeper.reportBug	1898
10.117.1	Function	1898
10.118	Tootsville.Game.Gatekeeper.rv	1899
10.118.1	Function	1899
10.118.2	See Also	1899
10.119	Tootsville.Game.Gatekeeper.scoreUpdate	1900
10.119.1	Function	1900
10.120	Tootsville.Game.Gatekeeper.sendMailMessage	1901
10.120.1	Function	1901
10.121	Tootsville.Game.Gatekeeper.serverTime	1902
10.121.1	Function	1902
10.122	Tootsville.Game.Gatekeeper.startEvent	1903
10.122.1	Function	1903
10.123	Tootsville.Game.Gatekeeper.tootList	1904
10.123.1	Function	1904
10.124	Tootsville.Game.Gatekeeper.wardrobe	1905
10.124.1	Function	1905
10.125	Tootsville.Game.Gatekeeper.wtl	1906
10.125.1	Function	1906
10.126	Tootsville.Game.GravitySystem.fastForward	1907
10.126.1	Function	1907
10.127	Tootsville.Game.GravitySystem.register	1908

10.127.1	Function	1908
10.128	Tootsville.Game.GravitySystem.updateEntityGravity	1909
10.128.1	Function	1909
10.129	Tootsville.Game.GravitySystem.updateGravity	1910
10.129.1	Function	1910
10.130	Tootsville.Game.GrowthSystem.evolve	1911
10.130.1	Function	1911
10.131	Tootsville.Game.GrowthSystem.fastForward	1912
10.131.1	Function	1912
10.132	Tootsville.Game.GrowthSystem.grow	1913
10.132.1	Function	1913
10.133	Tootsville.Game.GrowthSystem.register	1914
10.133.1	Function	1914
10.134	Tootsville.Game.GrowthSystem.remove	1915
10.134.1	Function	1915
10.135	Tootsville.Game.GrowthSystem.updateGrowth	1916
10.135.1	Function	1916
10.136	Tootsville.Game.MissileSystem.fastForward	1917
10.136.1	Function	1917
10.137	Tootsville.Game.MissileSystem.register	1918
10.137.1	Function	1918
10.138	Tootsville.Game.MissileSystem.remove	1919
10.138.1	Function	1919
10.139	Tootsville.Game.MissileSystem.updateMissiles	1920
10.139.1	Function	1920
10.140	Tootsville.Game.NPC.Collector.fastForward	1921
10.140.1	Function	1921
10.141	Tootsville.Game.NPC.Collector.register	1922
10.141.1	Function	1922
10.142	Tootsville.Game.NPC.Collector.updateNPC	1923
10.142.1	Function	1923
10.143	Tootsville.Game.NPC.Cook.fastForward	1924
10.143.1	Function	1924
10.144	Tootsville.Game.NPC.Cook.register	1925
10.144.1	Function	1925
10.145	Tootsville.Game.NPC.Cook.updateNPC	1926
10.145.1	Function	1926
10.146	Tootsville.Game.NPC.CroquetPlayer.fastForward	1927
10.146.1	Function	1927
10.147	Tootsville.Game.NPC.CroquetPlayer.register	1928
10.147.1	Function	1928
10.148	Tootsville.Game.NPC.CroquetPlayer.updateNPC	1929
10.148.1	Function	1929
10.149	Tootsville.Game.NPC.Doodle.fastForward	1930
10.149.1	Function	1930
10.150	Tootsville.Game.NPC.Doodle.register	1931
10.150.1	Function	1931
10.151	Tootsville.Game.NPC.Doodle.updateNPC	1932

10.151.1	Function	1932
10.152	Tootsville.Game.NPC.Fetcher.fastForward	1933
10.152.1	Function	1933
10.153	Tootsville.Game.NPC.Fetcher.register	1934
10.153.1	Function	1934
10.154	Tootsville.Game.NPC.Fetcher.updateNPC	1935
10.154.1	Function	1935
10.155	Tootsville.Game.NPC.JobWorker.fastForward	1936
10.155.1	Function	1936
10.156	Tootsville.Game.NPC.JobWorker.register	1937
10.156.1	Function	1937
10.157	Tootsville.Game.NPC.JobWorker.updateNPC	1938
10.157.1	Function	1938
10.158	Tootsville.Game.NPC.MazeBuilder.fastForward	1939
10.158.1	Function	1939
10.159	Tootsville.Game.NPC.MazeBuilder.register	1940
10.159.1	Function	1940
10.160	Tootsville.Game.NPC.MazeBuilder.updateNPC	1941
10.160.1	Function	1941
10.161	Tootsville.Game.NPC.Sleeper.fastForward	1942
10.161.1	Function	1942
10.162	Tootsville.Game.NPC.Sleeper.register	1943
10.162.1	Function	1943
10.163	Tootsville.Game.NPC.Sleeper.updateNPC	1944
10.163.1	Function	1944
10.164	Tootsville.Game.NPC.TrolleyDriver.fastForward	1945
10.164.1	Function	1945
10.165	Tootsville.Game.NPC.TrolleyDriver.register	1946
10.165.1	Function	1946
10.166	Tootsville.Game.NPC.TrolleyDriver.updateNPC	1947
10.166.1	Function	1947
10.167	Tootsville.Game.NPC.Waiter.fastForward	1948
10.167.1	Function	1948
10.168	Tootsville.Game.NPC.Waiter.register	1949
10.168.1	Function	1949
10.169	Tootsville.Game.NPC.Waiter.updateNPC	1950
10.169.1	Function	1950
10.170	Tootsville.Game.NPCSystem.burgeonNPC	1951
10.170.1	Function	1951
10.171	Tootsville.Game.NPCSystem.fastForward	1952
10.171.1	Function	1952
10.172	Tootsville.Game.NPCSystem.initNPCs	1953
10.172.1	Function	1953
10.172.2	NPC System Overview	1953
10.173	Tootsville.Game.NPCSystem.nextBehavior	1954
10.173.1	Function	1954
10.174	Tootsville.Game.NPCSystem.register	1955
10.174.1	Function	1955

10.175	Tootsville.Game.NPCSystem.updateNPC	1956
10.175.1	Function	1956
10.176	Tootsville.Game.NPCSystem.updateNPCs	1957
10.176.1	Function	1957
10.177	Tootsville.Game.Nav.CAMERA_MOVE_SPEED	1958
10.177.1	Variable	1958
10.178	Tootsville.Game.Nav.RUN_SPEED	1959
10.178.1	Variable	1959
10.179	Tootsville.Game.Nav.WALK_SPEED	1960
10.179.1	Variable	1960
10.180	Tootsville.Game.Nav.buildWTL	1961
10.180.1	Function	1961
10.181	Tootsville.Game.Nav.collisionP	1962
10.181.1	Function	1962
10.182	Tootsville.Game.Nav.mergeObjects	1963
10.182.1	Function	1963
10.183	Tootsville.Game.Nav.moveEntityOnCourse	1964
10.183.1	Function	1964
10.184	Tootsville.Game.Nav.quiesce	1965
10.184.1	Function	1965
10.185	Tootsville.Game.Nav.runTo	1966
10.185.1	Function	1966
10.186	Tootsville.Game.Nav.sendWTL	1967
10.186.1	Function	1967
10.187	Tootsville.Game.Nav.updateAvatar	1968
10.187.1	Function	1968
10.188	Tootsville.Game.Nav.updateAvatars	1969
10.188.1	Function	1969
10.189	Tootsville.Game.Nav.updateCamera	1970
10.189.1	Function	1970
10.190	Tootsville.Game.Nav.updateCameraDolly	1971
10.190.1	Function	1971
10.191	Tootsville.Game.Nav.updateCameraTruck	1972
10.191.1	Function	1972
10.192	Tootsville.Game.Nav.updateFacing	1973
10.192.1	Function	1973
10.193	Tootsville.Game.Nav.walkTheLine	1974
10.193.1	Function	1974
10.194	Tootsville.Game.Speech.createBalloon	1975
10.194.1	Function	1975
10.195	Tootsville.Game.Speech.dispatchCommand	1976
10.195.1	Function	1976
10.196	Tootsville.Game.Speech.removeSpeech	1977
10.196.1	Function	1977
10.197	Tootsville.Game.Speech.say	1978
10.197.1	Function	1978
10.198	Tootsville.Game.Speech.updateSpeech	1979
10.198.1	Function	1979

10.199	Tootsville.Game.Tools.axe	1980
10.199.1	Function	1980
10.200	Tootsville.Game.Tools.butterflyNet	1981
10.200.1	Function	1981
10.201	Tootsville.Game.Tools.fishingRod	1982
10.201.1	Function	1982
10.202	Tootsville.Game.Tools.pickaxe	1983
10.202.1	Function	1983
10.203	Tootsville.Game.Tools.sewingKit	1984
10.203.1	Function	1984
10.204	Tootsville.Game.Tools.shovel	1985
10.204.1	Function	1985
10.205	Tootsville.Game.Tools.wrench	1986
10.205.1	Function	1986
10.206	Tootsville.Game.Wardrobe	1987
10.206.1	Variable	1987
10.207	Tootsville.Game.Wardrobe.doff	1988
10.207.1	Function	1988
10.208	Tootsville.Game.Wardrobe.don	1989
10.208.1	Function	1989
10.209	Tootsville.Game.Wardrobe.drop	1990
10.209.1	Function	1990
10.210	Tootsville.Game.Wardrobe.finalizeExchange	1991
10.210.1	Function	1991
10.211	Tootsville.Game.Wardrobe.findBaseSlot	1992
10.211.1	Function	1992
10.212	Tootsville.Game.Wardrobe.inventory	1993
10.212.1	Function	1993
10.213	Tootsville.Game.Wardrobe.inventoryByKind	1994
10.213.1	Function	1994
10.214	Tootsville.Game.Wardrobe.proposeExchange	1995
10.214.1	Function	1995
10.215	Tootsville.Game.Wardrobe.readied	1996
10.215.1	Function	1996
10.216	Tootsville.Game.Wardrobe.readiedP	1997
10.216.1	Function	1997
10.217	Tootsville.Game.Wardrobe.ready	1998
10.217.1	Function	1998
10.218	Tootsville.Game.Wardrobe.refresh	1999
10.218.1	Function	1999
10.219	Tootsville.Game.Wardrobe.signExchange	2000
10.219.1	Function	2000
10.220	Tootsville.Game.Wardrobe.take	2001
10.220.1	Function	2001
10.221	Tootsville.Game.Wardrobe.wearing	2002
10.221.1	Function	2002
10.222	Tootsville.Game.Wardrobe.wearingP	2003
10.222.1	Function	2003

10.223	Tootsville.Game.bootstrap	2004
10.223.1	Function	2004
10.224	Tootsville.Game.clickedOnItem	2005
10.224.1	Function	2005
10.225	Tootsville.Game.credits	2006
10.225.1	Function	2006
10.226	Tootsville.Game.fastForward	2007
10.226.1	Function	2007
10.227	Tootsville.Game.hideWhenGameReady	2008
10.227.1	Function	2008
10.228	Tootsville.Game.interestingPoint	2009
10.228.1	Function	2009
10.229	Tootsville.Game.lag	2010
10.229.1	Variable	2010
10.230	Tootsville.Game.stopSlowLoadingWatchdogs	2011
10.230.1	Function	2011
10.231	Tootsville.Game.update	2012
10.231.1	Function	2012
10.232	Tootsville.Gossip.Parrot.ask	2013
10.232.1	Function	2013
10.233	Tootsville.Gossip.Parrot.done	2014
10.233.1	Function	2014
10.234	Tootsville.Gossip.Parrot.parrotErrorText	2015
10.234.1	Function	2015
10.235	Tootsville.Gossip.Parrot.say	2016
10.235.1	Function	2016
10.236	Tootsville.Gossip.Parrot.show	2017
10.236.1	Function	2017
10.237	Tootsville.Gossip.Parrot.ynP	2018
10.237.1	Function	2018
10.238	Tootsville.Gossip.acceptOffer	2019
10.238.1	Function	2019
10.239	Tootsville.Gossip.closeInfinityMode	2020
10.239.1	Function	2020
10.240	Tootsville.Gossip.closeStreams	2021
10.240.1	Function	2021
10.241	Tootsville.Gossip.connect	2022
10.241.1	Function	2022
10.242	Tootsville.Gossip.connectedP	2023
10.242.1	Function	2023
10.243	Tootsville.Gossip.createConnection	2024
10.243.1	Function	2024
10.244	Tootsville.Gossip.createPacket	2025
10.244.1	Function	2025
10.245	Tootsville.Gossip.ensureConnected	2026
10.245.1	Function	2026
10.246	Tootsville.Gossip.ensureKeyPair	2027
10.246.1	Function	2027



10.247	Tootsville.Gossip.gatekeeperAccept	2028
10.247.1	Function	2028
10.248	Tootsville.Gossip.getICE	2029
10.248.1	Function	2029
10.249	Tootsville.Gossip.getOffer	2030
10.249.1	Function	2030
10.250	Tootsville.Gossip.openInfinityMode	2031
10.250.1	Function	2031
10.251	Tootsville.Gossip.send	2032
10.251.1	Function	2032
10.252	Tootsville.Gossip.sendLogOK	2033
10.252.1	Function	2033
10.253	Tootsville.Gossip.signPacket	2034
10.253.1	Function	2034
10.254	Tootsville.Gossip.waitForAnswer	2035
10.254.1	Function	2035
10.255	Tootsville.GroundBuilder.build	2036
10.255.1	Function	2036
10.256	Tootsville.GroundBuilder.colorForPlace	2037
10.256.1	Function	2037
10.257	Tootsville.GroundBuilder.initGroundPlane	2038
10.257.1	Function	2038
10.258	Tootsville.GroundBuilder.kinds	2039
10.258.1	Variable	2039
10.259	Tootsville.GroundBuilder.paintPlaces	2040
10.259.1	Function	2040
10.260	Tootsville.Login.acceptSignedIn	2041
10.260.1	Function	2041
10.261	Tootsville.Login.addChildFlag	2042
10.261.1	Function	2042
10.262	Tootsville.Login.addChildRequest	2043
10.262.1	Function	2043
10.263	Tootsville.Login.changeSensitivePlayer	2044
10.263.1	Function	2044
10.264	Tootsville.Login.childRequestTimeLeft	2045
10.264.1	Function	2045
10.265	Tootsville.Login.childSettings	2046
10.265.1	Function	2046
10.266	Tootsville.Login.clearTootsList	2047
10.266.1	Function	2047
10.267	Tootsville.Login.considerChildApproval	2048
10.267.1	Function	2048
10.268	Tootsville.Login.createTootListItem	2049
10.268.1	Function	2049
10.269	Tootsville.Login.dimUnpickedCharacters	2050
10.269.1	Function	2050
10.270	Tootsville.Login.disableChildMode	2051
10.270.1	Function	2051

10.271	Tootsville.Login.doRealLogin	2052
10.271.1	Function	2052
10.272	Tootsville.Login.doneEditingSettings	2053
10.272.1	Function	2053
10.273	Tootsville.Login.enableChildMode	2054
10.273.1	Function	2054
10.274	Tootsville.Login.endLoginMusic	2055
10.274.1	Function	2055
10.275	Tootsville.Login.fillGoogleUserInfo	2056
10.275.1	Function	2056
10.276	Tootsville.Login.findLIForToot	2057
10.276.1	Function	2057
10.277	Tootsville.Login.finishSignIn	2058
10.277.1	Function	2058
10.278	Tootsville.Login.firebaseLogin	2059
10.278.1	Function	2059
10.279	Tootsville.Login.generateNewToot	2060
10.279.1	Function	2060
10.280	Tootsville.Login.loadTootsList	2061
10.280.1	Function	2061
10.281	Tootsville.Login.loginDone	2062
10.281.1	Function	2062
10.282	Tootsville.Login.loginKidDirty	2063
10.282.1	Function	2063
10.283	Tootsville.Login.loginKidDone	2064
10.283.1	Function	2064
10.284	Tootsville.Login.overlay	2065
10.284.1	Function	2065
10.285	Tootsville.Login.pickCharacter	2066
10.285.1	Function	2066
10.286	Tootsville.Login.playWithCharacter	2067
10.286.1	Function	2067
10.287	Tootsville.Login.populateTootsList	2068
10.287.1	Function	2068
10.288	Tootsville.Login.quit	2069
10.288.1	Function	2069
10.289	Tootsville.Login.removeChildFlag	2070
10.289.1	Function	2070
10.290	Tootsville.Login.saveTootsList	2071
10.290.1	Function	2071
10.291	Tootsville.Login.serverLinkTokenToCharacter	2072
10.291.1	Function	2072
10.292	Tootsville.Login.setSensitiveP	2073
10.292.1	Function	2073
10.293	Tootsville.Login.settingsP	2074
10.293.1	Variable	2074
10.294	Tootsville.Login.start	2075
10.294.1	Function	2075

10.295	Tootsville.Login.startCharacterCreation	2076
10.295.1	Function	2076
10.296	Tootsville.Login.startSignIn	2077
10.296.1	Function	2077
10.297	Tootsville.Login.storeCredentialInfo	2078
10.297.1	Function	2078
10.298	Tootsville.Login.switchTootsView	2079
10.298.1	Function	2079
10.299	Tootsville.Login.toots	2080
10.299.1	Variable	2080
10.300	Tootsville.Login.updateNote	2081
10.300.1	Function	2081
10.301	Tootsville.Login.validChildCode	2082
10.301.1	Function	2082
10.302	Tootsville.SceneBuilder.addFurn	2083
10.302.1	Function	2083
10.303	Tootsville.SceneBuilder.addItem1	2084
10.303.1	Function	2084
10.304	Tootsville.SceneBuilder.addItem2	2085
10.304.1	Function	2085
10.305	Tootsville.SceneBuilder.addPlace	2086
10.305.1	Function	2086
10.306	Tootsville.SceneBuilder.addText	2087
10.306.1	Function	2087
10.307	Tootsville.SceneBuilder.build	2088
10.307.1	Function	2088
10.308	Tootsville.SkyBuilder.build	2089
10.308.1	Function	2089
10.309	Tootsville.SkyBuilder.buildMatchingSky	2090
10.309.1	Function	2090
10.310	Tootsville.SkyBuilder.buildMatchingWeather	2091
10.310.1	Function	2091
10.311	Tootsville.SkyBuilder.setCloudCover	2092
10.311.1	Function	2092
10.312	Tootsville.SkyBuilder.setFirstSkyLayer	2093
10.312.1	Function	2093
10.313	Tootsville.SkyBuilder.setMoon	2094
10.313.1	Function	2094
10.314	Tootsville.SkyBuilder.setPlanet	2095
10.314.1	Function	2095
10.315	Tootsville.SkyBuilder.setPrecipitation	2096
10.315.1	Function	2096
10.316	Tootsville.SkyBuilder.setStarfield	2097
10.316.1	Function	2097
10.317	Tootsville.SkyBuilder.setSun	2098
10.317.1	Function	2098
10.318	Tootsville.SkyBuilder.setTheMoon	2099
10.318.1	Function	2099

10.319	Tootsville.SkyBuilder.setTheOtherMoon	2100
10.319.1	Function	2100
10.320	Tootsville.SkyBuilder.setThePinkMoon	2101
10.320.1	Function	2101
10.321	Tootsville.SkyBuilder.sunX	2102
10.321.1	Function	2102
10.322	Tootsville.SkyBuilder.sunY	2103
10.322.1	Function	2103
10.323	Tootsville.SkyBuilder.update	2104
10.323.1	Function	2104
10.324	Tootsville.SkyBuilder.updateSkyData	2105
10.324.1	Function	2105
10.325		
	Tootsville.Tank.CameraManager.positionCameraForAvatarCloseUp	2106
10.325.1	Function	2106
10.326		
	Tootsville.Tank.CameraManager.positionCameraForAvatarViewer	2107
10.326.1	Function	2107
10.327		
	Tootsville.Tank.CameraManager.positionCameraForGameBoard	2108
10.327.1	Function	2108
10.328	Tootsville.Tank.afterRender	2109
10.328.1	Function	2109
10.329	Tootsville.Tank.attachmentOverlaysNeedUpdateP	2110
10.329.1	Variable	2110
10.330	Tootsville.Tank.createScene	2111
10.330.1	Function	2111
10.331	Tootsville.Tank.destroyAvatar	2112
10.331.1	Function	2112
10.332	Tootsville.Tank.findAvatar	2113
10.332.1	Function	2113
10.333	Tootsville.Tank.getCanvas	2114
10.333.1	Function	2114
10.334	Tootsville.Tank.getLargestChildMesh	2115
10.334.1	Function	2115
10.335	Tootsville.Tank.init3DEngine	2116
10.335.1	Function	2116
10.336	Tootsville.Tank.initArcCamera	2117
10.336.1	Function	2117
10.337	Tootsville.Tank.initOTSCamera	2118
10.337.1	Function	2118
10.338	Tootsville.Tank.initPlayerToot	2119
10.338.1	Function	2119
10.339	Tootsville.Tank.initScene	2120
10.339.1	Function	2120
10.340	Tootsville.Tank.loadUISounds	2121
10.340.1	Function	2121
10.341	Tootsville.Tank.playerAvatar	2122

10.341.1	Function	2122
10.342	Tootsville.Tank.prepareFor3D	2123
10.342.1	Function	2123
10.343	Tootsville.Tank.shutDown	2124
10.343.1	Function	2124
10.344	Tootsville.Tank.start3D	2125
10.344.1	Function	2125
10.345	Tootsville.Tank.start3DReal	2126
10.345.1	Function	2126
10.346	Tootsville.Tank.startRenderLoop	2127
10.346.1	Function	2127
10.347	Tootsville.Tank.updateAvatarFor	2128
10.347.1	Function	2128
10.348	Tootsville.Tank.updateCamera	2129
10.348.1	Function	2129
10.349	Tootsville.UI.Audio.context	2130
10.349.1	Variable	2130
10.350	Tootsville.UI.Audio.gainNode	2131
10.350.1	Variable	2131
10.351	Tootsville.UI.Audio.setVolume	2132
10.351.1	Function	2132
10.352	Tootsville.UI.Audio.updateVolumeMuteIcon	2133
10.352.1	Function	2133
10.353	Tootsville.UI.Audio.updateVolumeSlider	2134
10.353.1	Function	2134
10.354	Tootsville.UI.Audio.updateVolumeUI	2135
10.354.1	Function	2135
10.355	Tootsville.UI.Audio.volumeDown	2136
10.355.1	Function	2136
10.356	Tootsville.UI.Audio.volumeMute	2137
10.356.1	Function	2137
10.357	Tootsville.UI.Audio.volumeUp	2138
10.357.1	Function	2138
10.358	Tootsville.UI.Gamepad.ROTATION_SPEED	2139
10.358.1	Variable	2139
10.359	Tootsville.UI.Gamepad.addGamepad	2140
10.359.1	Function	2140
10.360	Tootsville.UI.Gamepad.axisUpdate	2141
10.360.1	Function	2141
10.361	Tootsville.UI.Gamepad.buttonEvent	2142
10.361.1	Function	2142
10.362	Tootsville.UI.Gamepad.connectHandler	2143
10.362.1	Function	2143
10.363	Tootsville.UI.Gamepad.controllerState	2144
10.363.1	Variable	2144
10.364	Tootsville.UI.Gamepad.controllers	2145
10.364.1	Variable	2145
10.365	Tootsville.UI.Gamepad.disconnectHandler	2146

10.365.1	Function	2146
10.366	Tootsville.UI.Gamepad.removeGamepad	2147
10.366.1	Function	2147
10.367	Tootsville.UI.Gamepad.scanGamepads	2148
10.367.1	Function	2148
10.368	Tootsville.UI.Gamepad.updateStatus	2149
10.368.1	Function	2149
10.369	Tootsville.UI.HUD.beginWatchingPaperdollWindowForClose	2150
10.369.1	Function	2150
10.370	Tootsville.UI.HUD.clickedOnMesh	2151
10.370.1	Function	2151
10.371	Tootsville.UI.HUD.closePanel	2152
10.371.1	Function	2152
10.372	Tootsville.UI.HUD.closeTalkBox	2153
10.372.1	Function	2153
10.373	Tootsville.UI.HUD.connectTalkBox	2154
10.373.1	Function	2154
10.374	Tootsville.UI.HUD.convertCanvasEventTo3D	2155
10.374.1	Function	2155
10.375	Tootsville.UI.HUD.createHUDLoaderPanel	2156
10.375.1	Function	2156
10.376	Tootsville.UI.HUD.createPaperdollCanvas	2157
10.376.1	Function	2157
10.377	Tootsville.UI.HUD.destroyHUD	2158
10.377.1	Function	2158
10.378	Tootsville.UI.HUD.dropHUDPanels	2159
10.378.1	Function	2159
10.379	Tootsville.UI.HUD.getOpenPanel	2160
10.379.1	Function	2160
10.380	Tootsville.UI.HUD.initHUD	2161
10.380.1	Function	2161
10.381	Tootsville.UI.HUD.loadHTML	2162
10.381.1	Function	2162
10.382	Tootsville.UI.HUD.loadHUDPanel	2163
10.382.1	Function	2163
10.383	Tootsville.UI.HUD.loadScriptIntoDiv	2164
10.383.1	Function	2164
10.384	Tootsville.UI.HUD.nameTagClicked	2165
10.384.1	Function	2165
10.385	Tootsville.UI.HUD.openPaperdoll	2166
10.385.1	Function	2166
10.386	Tootsville.UI.HUD.openTalkBox	2167
10.386.1	Function	2167
10.387	Tootsville.UI.HUD.paperdollCurrentP	2168
10.387.1	Function	2168
10.388	Tootsville.UI.HUD.positionPaperdollMini	2169
10.388.1	Function	2169
10.389	Tootsville.UI.HUD.refreshAttachmentOverlays	2170

10.389.1	Function	2170
10.390	Tootsville.UI.HUD.refreshAttachmentsForAvatar	2171
10.390.1	Function	2171
10.391	Tootsville.UI.HUD.refreshEquipment	2172
10.391.1	Function	2172
10.392	Tootsville.UI.HUD.refreshHUD	2173
10.392.1	Function	2173
10.393	Tootsville.UI.HUD.refreshNameTagAttachment	2174
10.393.1	Function	2174
10.394	Tootsville.UI.HUD.refreshPaperdoll	2175
10.394.1	Function	2175
10.395	Tootsville.UI.HUD.refreshSpeechAttachment	2176
10.395.1	Function	2176
10.396	Tootsville.UI.HUD.refreshTalkStatus	2177
10.396.1	Function	2177
10.397	Tootsville.UI.HUD.refreshTimeLeft	2178
10.397.1	Function	2178
10.398	Tootsville.UI.HUD.refreshWallet	2179
10.398.1	Function	2179
10.399	Tootsville.UI.HUD.returnPaperdollMini	2180
10.399.1	Function	2180
10.400	Tootsville.UI.HUD.setPaperdollForPlayerAvatar	2181
10.400.1	Function	2181
10.401	Tootsville.UI.HUD.showCamera	2182
10.401.1	Function	2182
10.402	Tootsville.UI.HUD.showControlPanel	2183
10.402.1	Function	2183
10.403	Tootsville.UI.HUD.showHUDPanel	2184
10.403.1	Function	2184
10.404	Tootsville.UI.HUD.showMobile	2185
10.404.1	Function	2185
10.405	Tootsville.UI.HUD.showPlayerCard	2186
10.405.1	Function	2186
10.406	Tootsville.UI.HUD.switchActiveItem	2187
10.406.1	Function	2187
10.407	Tootsville.UI.HUD.talkBoxOpenP	2188
10.407.1	Variable	2188
10.408	Tootsville.UI.HUD.toggleElement	2189
10.408.1	Function	2189
10.409	Tootsville.UI.HUD.toggleHUDPanel	2190
10.409.1	Function	2190
10.410	Tootsville.UI.HUD.toggleTalkBox	2191
10.410.1	Function	2191
10.411	Tootsville.UI.HUD.toggleTalkEmoji	2192
10.411.1	Function	2192
10.412	Tootsville.UI.HUD.toggleTalkExpression	2193
10.412.1	Function	2193
10.413	Tootsville.UI.HUD.toggleTalkLoud	2194

10.413.1	Function	2194
10.414	Tootsville.UI.Keys.arrowDown	2195
10.414.1	Function	2195
10.415	Tootsville.UI.Keys.arrowLeft	2196
10.415.1	Function	2196
10.416	Tootsville.UI.Keys.arrowRight	2197
10.416.1	Function	2197
10.417	Tootsville.UI.Keys.arrowUp	2198
10.417.1	Function	2198
10.418	Tootsville.UI.Keys.backwardChar	2199
10.418.1	Function	2199
10.419	Tootsville.UI.Keys.backwardSentence	2200
10.419.1	Function	2200
10.420	Tootsville.UI.Keys.backwardWord	2201
10.420.1	Function	2201
10.421	Tootsville.UI.Keys.beginShouting	2202
10.421.1	Function	2202
10.422	Tootsville.UI.Keys.beginSpeaking	2203
10.422.1	Function	2203
10.423	Tootsville.UI.Keys.beginWhispering	2204
10.423.1	Function	2204
10.424	Tootsville.UI.Keys.beginningOfLine	2205
10.424.1	Function	2205
10.425	Tootsville.UI.Keys.capitalizeWord	2206
10.425.1	Function	2206
10.426	Tootsville.UI.Keys.deleteBackwardChar	2207
10.426.1	Function	2207
10.427	Tootsville.UI.Keys.deleteChar	2208
10.427.1	Function	2208
10.428	Tootsville.UI.Keys.downcaseWord	2209
10.428.1	Function	2209
10.429	Tootsville.UI.Keys.endOfLine	2210
10.429.1	Function	2210
10.430	Tootsville.UI.Keys.executeExtendedCommand	2211
10.430.1	Function	2211
10.431	Tootsville.UI.Keys.forwardChar	2212
10.431.1	Function	2212
10.432	Tootsville.UI.Keys.forwardSentence	2213
10.432.1	Function	2213
10.433	Tootsville.UI.Keys.forwardWord	2214
10.433.1	Function	2214
10.434	Tootsville.UI.Keys.help	2215
10.434.1	Function	2215
10.435	Tootsville.UI.Keys.insertChar	2216
10.435.1	Function	2216
10.436	Tootsville.UI.Keys.isearch	2217
10.436.1	Function	2217
10.437	Tootsville.UI.Keys.isearchBackward	2218



10.437.1	Function	2218
10.438	Tootsville.UI.Keys.keyboardQuit	2219
10.438.1	Function	2219
10.439	Tootsville.UI.Keys.killLine	2220
10.439.1	Function	2220
10.440	Tootsville.UI.Keys.killRegion	2221
10.440.1	Function	2221
10.441	Tootsville.UI.Keys.killRingSave	2222
10.441.1	Function	2222
10.442	Tootsville.UI.Keys.killSentence	2223
10.442.1	Function	2223
10.443	Tootsville.UI.Keys.killWord	2224
10.443.1	Function	2224
10.444	Tootsville.UI.Keys.nextHistoryLine	2225
10.444.1	Function	2225
10.445	Tootsville.UI.Keys.prefixCc	2226
10.445.1	Function	2226
10.446	Tootsville.UI.Keys.prefixCx	2227
10.446.1	Function	2227
10.447	Tootsville.UI.Keys.priorHistoryLine	2228
10.447.1	Function	2228
10.448	Tootsville.UI.Keys.selectAll	2229
10.448.1	Function	2229
10.449	Tootsville.UI.Keys.speakLine	2230
10.449.1	Function	2230
10.450	Tootsville.UI.Keys.textEntry	2231
10.450.1	Function	2231
10.451	Tootsville.UI.Keys.transposeChars	2232
10.451.1	Function	2232
10.452	Tootsville.UI.Keys.transposeWords	2233
10.452.1	Function	2233
10.453	Tootsville.UI.Keys.upcaseWord	2234
10.453.1	Function	2234
10.454	Tootsville.UI.Keys.yank	2235
10.454.1	Function	2235
10.455	Tootsville.UI.Keys.yankPop	2236
10.455.1	Function	2236
10.456	Tootsville.UI.WaWa.build	2237
10.456.1	Function	2237
10.457	Tootsville.UI.WaWa.playChained	2238
10.457.1	Function	2238
10.458	Tootsville.UI.WaWa.playShifted	2239
10.458.1	Function	2239
10.459	Tootsville.UI.WaWa.stop	2240
10.459.1	Function	2240
10.460	Tootsville.UI.clickedOnItem	2241
10.460.1	Function	2241
10.461	Tootsville.UI.commands	2242

10.461.1	Variable	2242
10.462	Tootsville.UI.confirmPretty	2243
10.462.1	Function	2243
10.463	Tootsville.UI.findAdjacentEntity	2244
10.463.1	Function	2244
10.464	Tootsville.UI.forceQuit	2245
10.464.1	Function	2245
10.465	Tootsville.UI.insertEmoji	2246
10.465.1	Function	2246
10.466	Tootsville.UI.interact	2247
10.466.1	Function	2247
10.467	Tootsville.UI.makeDivOrParagraph	2248
10.467.1	Function	2248
10.468	Tootsville.UI.makeIDFromTitle	2249
10.468.1	Function	2249
10.469	Tootsville.UI.makePrettyDialog	2250
10.469.1	Function	2250
10.470	Tootsville.UI.makePrompt	2251
10.470.1	Function	2251
10.471	Tootsville.UI.onFirstClick	2252
10.471.1	Function	2252
10.472	Tootsville.UI.quit	2253
10.472.1	Function	2253
10.473	Tootsville.UI.runCommand	2254
10.473.1	Function	2254
10.474	Tootsville.UI.say	2255
10.474.1	Function	2255
10.475	Tootsville.UI.setFullscreen	2256
10.475.1	Function	2256
10.476	Tootsville.UI.setFullscreenFromNavigator	2257
10.476.1	Function	2257
10.477	Tootsville.UI.signOut	2258
10.477.1	Function	2258
10.478	Tootsville.UI.slowLoadingWatchdog	2259
10.478.1	Function	2259
10.479	Tootsville.UI.takeOneStep	2260
10.479.1	Function	2260
10.480	Tootsville.UI.toggleFullscreen	2261
10.480.1	Function	2261
10.481	Tootsville.UI.useActiveItem	2262
10.481.1	Function	2262
10.482	Tootsville.Util.assertValidHostName	2263
10.482.1	Function	2263
10.483	Tootsville.Util.ensureServersReachable	2264
10.483.1	Function	2264
10.484	Tootsville.Util.equalP	2265
10.484.1	Function	2265
10.485	Tootsville.Util.infinityAwaits	2266

10.485.1	Function	2266
10.486	Tootsville.Util.loadScript	2267
10.486.1	Function	2267
10.487	Tootsville.Util.rest	2268
10.487.1	Function	2268
10.488	Tootsville.decodeTime	2269
10.488.1	Function	2269
10.489	Tootsville.gamepadLayouts	2270
10.489.1	Variable	2270
10.490	Tootsville.host	2271
10.490.1	Variable	2271
10.491	Tootsville.universalTimeOffset	2272
10.491.1	Variable	2272
10.492	Tootsville.updateClock	2273
10.492.1	Function	2273
10.493	window.interpretTootColor	2274
10.493.1	Function	2274
10.494	window.onGoogleYoloLoad	2275
10.494.1	Function	2275
<b>11</b>	<b>Credits</b>	<b>2277</b>
11.1	Major Support Software	2277
11.2	Systems	2278
11.2.1	System Tootsville	2278
11.2.2	System Twilio	2278
11.2.3	System Thread-Pool-Taskmaster	2278
11.2.4	System Verbose	2278
11.2.5	System Documentation-Utils	2278
11.2.6	System Trivial-Indent	2278
11.2.7	System Dissect	2278
11.2.8	System Local-Time	2279
11.2.9	System Piping	2280
11.2.10	System Alexandria	2280
11.2.11	System Rollbar	2280
11.2.12	System Oliphaunt	2280
11.2.13	System Usocket	2280
11.2.14	System Sb-Bsd-Sockets	2280
11.2.15	System Trivial-Gray-Streams	2280
11.2.16	System Trivial-Garbage	2281
11.2.17	System St-Json	2281
11.2.18	System Sqlite	2281
11.2.19	System Iterate	2281
11.2.20	System Split-Sequence	2281
11.2.21	System Prepl	2281
11.2.22	System Named-Readtables	2282
11.2.23	System Conium	2282
11.2.24	System Closer-Mop	2282
11.2.25	System Parse-Number	2282

11.2.26	System Langutils	2282
11.2.27	System Stdutils	2282
11.2.28	System S-Xml-Rpc	2282
11.2.29	System S-Xml	2282
11.2.30	System Cffi	2283
11.2.31	System Babel	2283
11.2.32	System Trivial-Features	2283
11.2.33	System Cl-Unicode	2283
11.2.34	System Cl-Unicode/ Base	2283
11.2.35	System Cl-Readline	2283
11.2.36	System Cl-Oauth	2283
11.2.37	System Puri	2283
11.2.38	System F-Underscore	2283
11.2.39	System Anaphora	2284
11.2.40	System Ironclad	2284
11.2.41	System Sb-Posix	2284
11.2.42	System Sb-Rotate-Byte	2284
11.2.43	System Cl-Fad	2284
11.2.44	System Buildapp	2284
11.2.45	System Apply-Argv	2284
11.2.46	System Dreamhost	2284
11.2.47	System Uuid	2284
11.2.48	System Trivial-Utf-8	2285
11.2.49	System Uiop	2285
11.2.50	System Trivial-Signal	2285
11.2.51	System Trivial-Ldap	2285
11.2.52	System Yacc	2285
11.2.53	System Cl+Ssl	2286
11.2.54	System Flexi-Streams	2286
11.2.55	System Trivial-Backtrace	2286
11.2.56	System Symbol-Munger	2287
11.2.57	System Swank	2287
11.2.58	System Pngload	2287
11.2.59	System Zpb-Exif	2287
11.2.60	System Swap-Bytes	2287
11.2.61	System Parse-Float	2287
11.2.62	System 3bz	2288
11.2.63	System Nibbles	2288
11.2.64	System Lparallel	2288
11.2.65	System Jonathan	2288
11.2.66	System Cl-Annot	2288
11.2.67	System Proc-Parse	2288
11.2.68	System Sb-Cltl2	2288
11.2.69	System Trivial-Types	2288
11.2.70	System Fast-IO	2288
11.2.71	System Static-Vectors	2289
11.2.72	System Cl-Syntax-Annot	2289
11.2.73	System Cl-Syntax	2289

11.2.74	System Hunchensocket .....	2289
11.2.75	System Chunga .....	2290
11.2.76	System Hunchentoot .....	2290
11.2.77	System Rfc2388 .....	2290
11.2.78	System Md5 .....	2290
11.2.79	System Fare-Memoization .....	2293
11.2.80	System Envy .....	2293
11.2.81	System Drakma .....	2293
11.2.82	System Chipz .....	2293
11.2.83	System Dbd-Mysql .....	2293
11.2.84	System Cl-Mysql .....	2293
11.2.85	System Dbi .....	2294
11.2.86	System Darts.Lib.Email-Address .....	2294
11.2.87	System Cxml .....	2294
11.2.88	System Cxml/ Klacks .....	2305
11.2.89	System Cxml/ Xml .....	2317
11.2.90	System Closure-Common .....	2328
11.2.91	System Cxml/ Dom .....	2328
11.2.92	System Clouddb .....	2340
11.2.93	System S-Base64 .....	2340
11.2.94	System Parensript .....	2340
11.2.95	System Cljwt-Custom .....	2341
11.2.96	System Yason .....	2341
11.2.97	System Cl-Smtp .....	2341
11.2.98	System Cl-Ppcre .....	2341
11.2.99	System Cl-Memcached .....	2341
11.2.100	System Pooler .....	2342
11.2.101	System Sb-Concurrency .....	2342
11.2.102	System Cl-Dbi .....	2342
11.2.103	System Cl-Base64 .....	2342
11.2.104	System Bordeaux-Threads .....	2343
11.2.105	System Global-Vars .....	2343
11.3	The Steel Bank Common Lisp compiler .....	2343
11.4	Javascript Tools .....	2361
<b>12</b>	<b>Conclusion .....</b>	<b>2363</b>
12.1	License .....	2363
12.1.1	Tootsville Contents .....	2363
12.1.2	Additional Media Content .....	2363
12.2	AGPL v3 License .....	2363
12.2.1	GNU AFFERO GENERAL PUBLIC LICENSE .....	2363
<b>Appendix A</b>	<b>Indices .....</b>	<b>2375</b>
A.1	Concepts .....	2375
A.2	Functions .....	2376
A.3	Variables .....	2395
A.4	Data types .....	2397

A.5	Pathnames .....	2399
A.6	Infinity Mode commands.....	2402
A.7	Operator commands .....	2404
A.8	Game Actions .....	2406
A.9	Javascript .....	2407

## Copying

This program is free software; you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License along with this program (one is found in this book); if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.





# 1 Introduction

This is the manual for the tootsville server for CIWTA.

## 1.1 Who are CIWTA?

CIWTA, The Corporation for Inter-World Tourism and Adventuring, is the non-profit corporation responsible for the current development of the Romance Game System, particularly, for the flagship implementation of it in the form of Tootsville .

For more information about CIWTA, visit the web site at <https://www.ciwta.org/>

## 1.2 What is Tootsville?

Tootsville is a massively-multiplayer online rôle-playing game (MMORPG) in which players control colorful, elephant-like characters know as Toots.

Tootsville was the flagship product which helped develop the original Romance game system into such a powerful suite. Its commercial sponsor, Res Interactive, LLC, was later shut down for legal reasons. The resurrected game (with the permission of several of Res’s managing members) is now the flagship for the ongoing development of Romance .

As such, Romance is highly and unapologetically targeted at present towards the needs of Tootsville.

## 1.3 What is the Romance Game System?

Romance is the name of the game system which runs Tootsville, and which could, in future, be adapted to run other games as well.

The name “Romance” has a similar meaning to its usage in “Romance Languages,” i.e. it means made of Romans, not “romantic.” The original software modules of which Romance was made, and into which it will be subdivided again for purposes of modularity in the 2.0 series, are each named after a famous Roman.

Romance has a few technological goals. It is intended to be comprehensive and flexible enough to handle a variety of game worlds, although it necessarily has deep ties to Tootsville. Romance also is based on a model in which peer-to-peer communications are used for real time activities, while the central servers are reserved for persisting the world’s state while it is quiescent.

## 1.4 Technology Stack

Romance is built up of the following technology stack:

- The central REST servers are written in Common Lisp, compiled to a native binary, and run as a service under SystemD in a non-privileged user account.
- The client and peer-to-peer application is written in JavaScript (ECMAScript 6 level), and compressed using the Google Closure Compiler into an optimized and minified form.

## 1.5 Affiliated Services

In the case of Tootsville, several affiliated services are used:

- Static web servers running Apache serve up game assets, compiled JavaScript, and other resources.
- Apache servers operate as load balancers across the back-end game servers
- The Tootsbook blog is a WordPress installation
- The central database server is a MariaDB server.
- Database references are cached via MemCached

## 1.6 Clusters

There are four cluster types for Tootsville.

`local` This refers to running a full Tootsville stack on a developer's workstation

`test.Tootsville.org`

This cluster is used for testing new builds and is the most unstable, often having a new release pushed to it every week — sometimes, several in one day.

`qa.Tootsville.org`

This cluster is used for longer-term testing of the code before rolling it out to the general public.

`Tootsville.org`

This is the main production environment

## 1.7 Overview of Major Systems

There are several major systems worth understanding from a higher level. Most of these are documented under specific functions that are critical in those subsystems.

### 1.7.1 Methods of Connecting

#### 1.7.1.1 REST Requests

REST calls can be anonymous (or public), or require third-party authentication, i.e. Firebase credentials.

WRITEME

#### 1.7.1.2 Infinity Mode communications

In the beginning, Tootsville I, the Hillside Demo, there was SmartFox Server. This was a chat server designed to work over an XML protocol with Adobe Flash clients. Tootsville I was built on this SmartFox Server and the SmartFox client software that went with it.

Unfortunately, SFS was not able to scale up with Tootsville's growth, and was very resource-intensive on the server side, so Bruce-Robert Pocock, the Chief Engineer at Res Interactive, brought in a Java-based chat server that he had written, named Braque. Braque was renamed Appius Claudius Caecus, and became the first Roman of the Romance Game System.

In order to convince the SmartFox Client software in the Tootsville Flosch client program (first Nightmare, and later Persephone) to communicate with Appius, we had to advertise a SmartFox version number — so, in order to ensure that we had a sufficiently high version number, and since EcmaScript uses floating-point numbers, we chose Infinity.

The protocol gradually turned into a JSON-oriented library of functions, leaving behind the SFS protocols (although some SFS concepts remained, such as room variables and user variables, in various forms).

The modern version of Infinity Protocol over WebSockets and TCP streaming is known as version Alef-Null, which is a fascinating maths concept that refers to a certain kind of Infinity.

There are 3 types of authentication supported for Infinity mode: Adult Sign-In, Child Sign-In, and Server-to-Server.

Before authenticating, a very limited vocabulary is available; see Section 8.719 [TOOTSVILLE INFINITY-PRE-LOGIN], page 990, for a discussion of what is available to end users. Server-to-server connections send their authentication in advance.

Once authenticated, the vocabulary grows extensively. See Section 8.338 [TOOTSVILLE DEFINFINITY], page 592, for an overview of Infinity Mode commands and how they can also be called as REST endpoints. Commands begin with INFINITY-, and can be found alphabetically in Chapter 2 [Definitions], page 11.

Note that some of these are deprecated or no longer useful, but all commands since 1.0 are still included in the vocabulary, including some which were originally Res Interactive proprietary extensions.

See Appendix 6 for an index of Infinity Mode commands.

### 1.7.1.3 The Adult Sign-in Process

Adults signing in must be first authenticated by an outside provider. Presently we're using Firebase for that purpose. Long-term, post-5.0 we'd like to switch to handling pure OAuth on our own, but migrating from Firebase could be problematic and take some time to phase in, so we'd prefer to do that before we reach too large a critical mass of users.

Once the client has performed authentication and received its tokens, it will send a login packet that is authenticated by Section 8.1346 [TOOTSVILLE WEBSOCKET-AUTHENTICATE], page 1633, and Section 8.586 [TOOTSVILLE FIND-USER-FOR-JSON], page 843.

WRITEME

### 1.7.1.4 The Child Sign-in Process

Child sign-ins are conducted using the Section 8.693 [TOOTSVILLE INFINITY-GET-APPLE], page 956, function.

WRITEME

### 1.7.1.5 The Server-to-Server Sign-In Process

Server-to-Server peering occurs on TCP port 5005, and is accessible only via the ::1 (loopback) interface. This means that, in order to create a server-to-server peering, the connection must be tunneled over ssh first. Since this is an unattended server process, this requires the use of public/private key pairs to establish trust between servers.

The server-to-server stream uses JSON packets packaged into a simple ANSI-control-characters-based streaming protocol.

Sign-in occurs as follows . . . WRITEME

WRITEME

## 1.7.2 In-Game Actions

### 1.7.2.1 Moving in the Game

There have been 3 systems for character movement.

The oldest dates back to Tootsville I and was the `d` method. This is no longer in use. For documentation, to the extent any exists, refer to Tootsville IV docs. In brief, it allowed a Toot to do basically the same thing as the `wtl` method, only it compacted the description into a string joined with `~` characters.

The current *status quō* method is the “d” method, or “walk the line” (`wtl`) method. The basic concept is that each Toot’s position is determined by a linear interpolation along a straight line described by a start and end position, a speed of movement, and a start time. Thus, all clients should be able to reliably place a character at the same point on the line, regardless of any lag in the transmission. See Section 8.744 [TOOTSVILLE INFINITY-WTL], page 1028- for a discussion of this method.

A more complex system being designed for post-5.0 use is the “d” method, which allows the client to perform pathfinding and create a Bezier spline walk pattern. This system is loosely supported by the server but in non-specific ways.

### 1.7.2.2 Speech and Related Things

Speech mostly consists of public messages. Each public message contains a volume level, speech contents, and musical key (for the Toot sounds). Section 8.736 [TOOTSVILLE INFINITY-SPEAK], page 1016, handles the bulk of speech.

Private messaging is accomplished by whispering to another player using `@` messages — i.e. the message begins with `@` and the other player’s name, and is processed by the server as a special whisper command.

Operator (Builder Toot) commands begin with a `#` and are processed by the server. See Appendix 7 for an index of operator commands.

Client-side commands begin with a `~` and are processed by the client, without ever sending them to the server.

### 1.7.2.3 Game Events System (including Store Items)

WRITEME

See Section 8.738 [TOOTSVILLE INFINITY-START-EVENT], page 1020, and Section 8.689 [TOOTSVILLE INFINITY-END-EVENT], page 949, for an overview.

### 1.7.2.4 Land Ownership

WRITEME

### 1.7.2.5 Clothing, Tools, and Equipment

Clothing, Tools, and Equipment are “just” items which happen to be able to be held in a player’s inventory. This is largely a function of the weight assigned and the carrying capacity of the character.

These items feature a Wear Slot value. A Wear Slot indicates a point on an avatar at which a piece of clothing can be mounted, or an item can be held. These slots are distinct to an avatar type, so UltraToot has different Wear Slots than, say, Jack or Welduh.

Wear Slots have valences that allow multiple layers of clothing to occupy the same essential slot: eg, a T-shirt under a blazer. Some articles of clothing may be defined to block other slots or other valences; eg a full-length dress might block a shirt or pants both.

Items have energy, which can be measured in a continuous or discrete way. Continuous energy types are effectively a continuum of rational values, and energy can be expended in any fraction of that amount. Discrete energy types are an integer counter, and a specific count is displayed to the user. When an item’s energy reaches zero, it can vanish, or just remain in inventory awaiting a recharge.

Equipment and tools have special hooks to enable them to be “used” in the game world. First, they must be held in the player’s trunk (for avatars with hands, they can be in the left or right hand). Second, there must be a “power” associated with the item, which requires a client-side function specialized on the item’s template ID. This hook may be a simple wrapper around reporting back to the server, or it can be as ornate as necessary.

WRITEME

### 1.7.2.6 Metronome

The metronome system allows tasks to occur on a recurring basis without having to keep their own timing threads open all the time. It also provides for one-shot events to run at a specific future time.

The main point of entry for scheduling a Metronome task is DO-METRONOME. The metronome thread itself relies upon RUN-METRONOME-TASKS to actually start tasks on each cycle.

Programmers are strongly encouraged to schedule tasks using Metronome throughout the game code.

## 1.7.3 World Simulation

WRITEME

## 1.7.4 Server-to-Server Streams

WRITEME

## 1.7.5 The front-end

The front-end services of Tootsville are provided by a JavaScript program, in the repository <https://github.com/adventuring/tootsville.org>.

### 1.7.5.1 Coding Standard

In general, the following coding standards apply to the front-end:

- Everything should be in the global `Tootsville` object-as-namespace.

- Within that object, each subsystem has its own object-as-namespace.
- Namespaces and classes use CamelCase; functions, variables, and object property names use lowerCamelCase names.
- When defining any object in the global namespace, it must be defined in such a way that reloading the file is idempotent, and it must not erase any other members that may have been added into the same object-as-namespace.

The effects of this is that a typical Javascript source file will need to contain a series of declarations like this:

```
if (!('Tootsville' in window))
{ Tootsville = {}; }

if (!('Namespace' in Tootsville))
{ Tootsville.Namespace = {}; }

Tootsville.Namespace.funcName =
  function (lambda, list)
  { ... };

Tootsville.Namespace.object =
  { foo: 42 };
```

Note, in particular, that we *must not* do something like:

```
Tootsville.Namespace =
  { funcName: function () { ... } };
```

This would potentially remove other objects in the `Tootsville.Namespace` namespace that may have been defined by other users.

### 1.7.5.2 Babylon.js

The front-end's 3D support is courtesy of the Babylon.js library, which has its own on-line documentation.

### 1.7.5.3 Gatekeeper

The Gatekeeper object contains the bulk of the client's command processing. Functions in Gatekeeper are named – i.e. the keys in the Gatekeeper hash table – for the datagram `from` keys sent by the server. For example, `Tootsville.Game.Gatekeeper.wt1` is the handler for datagrams with `from: "wt1"` in their packet.

WRITE ME

### 1.7.5.4 Peer-to-Peer Streams (WebRTC)

WRITE ME

### 1.7.5.5 JSCL

WRITE ME

## 1.8 Back Story

Once upon a time — for all fairy stories begin once upon a time — but not too very long ago, there were a herd of elephants wandering the plains of Africa. These elephants were just “ordinary” elephants, but of course each of them had their own personality.

...

WRITE ME

...

### 1.8.1 The Magic Mist and Mist Parrots

The elephants discover the magic mist ... parrots guide them through ...

WRITE ME

### 1.8.2 The Founding of Tootsville

Discover the fountain ... wish things into being ...

WRITE ME

### 1.8.3 The Classical Period

For many years, the Toots expanded their domain across the south and west of Tootanga.

WRITE ME

### 1.8.4 The Evil Mayor and Shade

All was well and good, until the arrival of Shade. Shade was a monster made up of black and purple smoke, with glowing orange eyes, created from bad wishes — any wishes that weren't for good. Shade and his team of Shaddows — Welduh, Smudge, Nevermind, and others — arrived on the scene and started making trouble for the Toots.

Some Toots even joined up with Shade and his Shaddows, and traveled to the evil valley of Shaddowfalls in the northeastern mountains.

Zap and the others used their power to keep Shade at bay, although from time to time Shade won small victories, at one time even changing all of Tootsville into Shadesville for a couple of weeks.

To fight the ongoing perils of Shade, the Toot Troops were organized, a scouting-type organization that kept the Shaddows on notice.

Then, Shade came up with his greatest plan — he replaced the mayor of Tootsville with the evil mayor, a hollow Toot full of smoke and stuffed with straw. The evil mayor helped Shade get into the underground Troops bunker and disconnect the water leading to the Toot Square Fountain. Smudge went around and poisoned all of the other wishing fountains, leaving the Toots without any source of Wish Magic. Ogres took control of the Enchanted Forest and kept the Toot Fairies from bringing in any Fairy Dust.

### 1.8.5 The Destruction of Tootsville

Things were bleak. Tootsville fell, and the Toots one by one disappeared, leaving behind nothing but a field of purple volcanic dust and Shade.

### 1.8.6 The Revival

Not everyone was destroyed. Pil, the most powerful Toot of all, protected some of the other Toots from Shade. Codfish-Howie and Catvle got together and revived the original 8 Toots, then more and more. They trapped Shade in a forcefield of his own evil magic in Shaddowfalls, and began to rebuild Tootsville, starting with the iconic Toot Square Fountain.

That brings us to today.



## 2 Definitions

The following chapters provide documentation of symbols in each of the main packages used by Tootsville. Many other libraries are relied upon as well, whose documentation may not have been included here.



### 3 Package Choerogryllum

## **3.1 Choerogryllum::Cal-Month**

### **3.1.1 Function**

Cal-Month names a function, with lambda list (YEAR MONTH):

Pretty-prints a one-month mini-calendar.

### **3.1.2 File**

Defined in file `src/lib/Choerogryllum/Choerogryllum.lisp`.

## 3.2 Chœrogyllum::Cal-Month-Header

### 3.2.1 Function

Cal-Month-Header names a function, with lambda list (YEAR MONTH STREAM):

Prints a header for a calendar of MONTH in YEAR to STREAM.

### 3.2.2 File

Defined in file `src/lib/Chœrogyllum/Chœrogyllum.lisp`.

### **3.3 Choerogryllum::Cal-Month-Header.Html**

#### **3.3.1 Function**

Cal-Month-Header.Html names a function, with lambda list (YEAR MONTH STREAM):

Writes an HTML header for a calendar of MONTH in YEAR to STREAM.

#### **3.3.2 File**

Defined in file src/lib/Choerogryllum/Choerogryllum.lisp.

## 3.4 Chœrogyllum::Cal-Month.Html

### 3.4.1 Function

Cal-Month.Html names a function, with lambda list (&OPTIONAL (YEAR (THIS-YEAR)) (MONTH (THIS-MONTH))):

Pretty-prints a one-month mini-calendar.

### 3.4.2 File

Defined in file src/lib/Chœrogyllum/Chœrogyllum.lisp.

## **3.5 Chœrogryllum::Cal-Month/ Print-Holiday-Footnotes**

### **3.5.1 Function**

Cal-Month/Print-Holiday-Footnotes names an undocumented function, with lambda list (YEAR MONTH HOLIDAYS STREAM).

### **3.5.2 File**

Defined in file src/lib/Chœrogryllum/Chœrogryllum.lisp.



## **3.6 Chœrogryllum::Cal-Year**

### **3.6.1 Function**

Cal-Year names an undocumented function, with lambda list (YEAR).

### **3.6.2 File**

Defined in file `src/lib/Chœrogryllum/Chœrogryllum.lisp`.

## 3.7 Choerogryllum::Date-String

### 3.7.1 Function

Date-String names a function, with lambda list (TIME &KEY (FORM LONG)):

Returns the pretty-printed Choerogryllum date string describing Universal time TIME.

### 3.7.2 File

Defined in file `src/lib/Choerogryllum/Choerogryllum.lisp`.

## 3.8 Chœrogyllum::Day-Of-Week\*

### 3.8.1 Function

Day-Of-Week\* names an undocumented function, with lambda list (I &KEY (FORM LONG)).

### 3.8.2 File

Defined in file src/lib/Chœrogyllum/Chœrogyllum.lisp.

## 3.9 Chœrogryllum::Decode\*-Universal-Time

### 3.9.1 Function

Decode\*-Universal-Time names a function, with lambda list (&OPTIONAL (TIME (GET-UNIVERSAL-TIME))):

Returns multiple values with date and time decoded.

Returns: (sec min hour day month year weekday other-month-day pink-month-day julian)

### 3.9.2 File

Defined in file src/lib/Chœrogryllum/Chœrogryllum.lisp.

## 3.10 Chœrogyllum::Encode\*-Universal-Time

### 3.10.1 Function

Encode\*-Universal-Time names a function, with lambda list (SEC MIN HOUR DAY MONTH YEAR):

Encodes a Chœrogyllum date & time into a Universal Time.

### 3.10.2 File

Defined in file src/lib/Chœrogyllum/Chœrogyllum.lisp.

## **3.11 Chœrogryllum::Exponent-Digit**

### **3.11.1 Function**

Exponent-Digit names a function, with lambda list (NUMBER):

Returns the digit NUMBER in exponent (superscript) character form

### **3.11.2 File**

Defined in file `src/lib/Chœrogryllum/Chœrogryllum.lisp`.

## 3.12 Choerogryllum::First-Weekday-Of-Month

### 3.12.1 Function

First-Weekday-Of-Month names a function, with lambda list (YEAR MONTH):

Returns the weekday number (0-8) of the first day of MONTH in YEAR.

### 3.12.2 File

Defined in file `src/lib/Choerogryllum/Choerogryllum.lisp`.

### 3.13 Chœrogyllum::Holiday-On

#### 3.13.1 Function

Holiday-On names a function, with lambda list (YEAR MONTH DAY):

Returns the name of any holiday on YEAR, MONTH, DAY.

YEAR, MONTH, and DAY are the integral values of the Chœrogyllym year, month, and day.

If there is no holiday, but there is a full moon (any moon), that may be reported instead.

#### 3.13.2 Chœrogyllum Holiday

The following holidays are recognized and reported:

**Trimestus** occurs when all three moons are full. Since the months are evenly matched to the phases of The Moon, this will always occur on the 15 day of some month. This is a major festival day.

**Hallowe'en**

occurs on the 30 day of Procavia (month 10).

**Hallowsday**

occurs on the 1st day of Dendrohyrax (month 11).

**Easter**

occurs on Lightningsday some time between Inunguis, Manatus, or Hydrodamalis, (months 3-5) based on the phase of The Other Moon.

**Christmas**

occurs on the 25 day of Tethytheria (month 12).

**Christmas Eve**

occurs on the 24 day of Tethytheria (month 12).

**Parents' Day**

occurs on the 13 day of Hydrodamalis (month 5). It takes the place of both Mothers' Day and Fathers' Day on Earth.

**The Winter Solstice**

occurs on the 21 day of Sirenia (month 1).

**The Spring Equinox**

occurs on the 21 day of Manatus (month 4).

**The Summer Solstice**

occurs on the 21 day of Pygmaeus (month 7).

**The Autumn Equinox**

occurs on the 21 day of Procavia (month 10).

**Fawkesday**

occurs on the 5 day of Dendrohyrax (month 11).

**New Year's Day**

occurs on the first day of Sirenia (month 1).

**The Summer Arts Festival**

runs from the 17th to the 20th day of Pygmaeus, except on Blanksday.



**Duomestus**

occurs whenever the Other Moon and Pink Moon are both full

**3.13.3 File**

Defined in file `src/lib/Chœrogryllum/Chœrogryllum.lisp`.

## 3.14 Chœrogryllum::Month\*

### 3.14.1 Function

Month\* names an undocumented function, with lambda list (I &KEY (FORM LONG)).

### 3.14.2 File

Defined in file src/lib/Chœrogryllum/Chœrogryllum.lisp.

## **3.15 Chœrogyllum::This-Month**

### **3.15.1 Function**

This-Month names an undocumented function, with lambda list NIL.

### **3.15.2 File**

Defined in file `src/lib/Chœrogyllum/Chœrogyllum.lisp`.

## **3.16 Chœrogryllum::This-Year**

### **3.16.1 Function**

This-Year names an undocumented function, with lambda list NIL.

### **3.16.2 File**

Defined in file `src/lib/Chœrogryllum/Chœrogryllum.lisp`.

## 4 Package Dreamhost

## 4.1 Dreamhost::**\*Api-Key\***

### 4.1.1 Variable

**\*Api-Key\*** names a variable:

The Dreamhost API Key to be used.

Generate one in the Panel at <https://panel.dreamhost.com/?tree=home.api>

Its value is NIL

## 4.2 Dreamhost::Cname-Already-On-Record

### 4.2.1 Class

Cname-Already-On-Record names a class, with one superclass: Section 4.7 [DREAMHOST DREAMHOST-API-ERROR], page 38.

### 4.2.2 Slots

Class Cname-Already-On-Record has no direct slots defined.

## **4.3 Dreamhost::Cname-Must-Be-Only-Record**

### **4.3.1 Class**

Cname-Must-Be-Only-Record names a class, with one superclass: Section 4.7 [DREAMHOST DREAMHOST-API-ERROR], page 38.

### **4.3.2 Slots**

Class Cname-Must-Be-Only-Record has no direct slots defined.



## 4.4 Dreamhost::Dns-Add-Record

### 4.4.1 Function

Dns-Add-Record names a function, with lambda list (NAME TYPE &OPTIONAL VALUE COMMENT):

Add a DNS record for NAME of TYPE; VALUE is determined by TYPE.

Adds a new DNS record to a domain you already have hosted with DreamHost. However, you cannot add dreamhosters.com records. Keep in mind DNS changes may take a while to propagate.

**type:** A, CNAME, NS, PTR, NAPTR, SRV, TXT, or AAAA

### 4.4.2 Result success

record\_added

### 4.4.3 Possible Errors

- no\_record
- no\_type
- no\_value
- invalid\_record (may have specifics after a tab)
- invalid\_type (may have specifics after a tab)
- invalid\_value (may have specifics after a tab)
- no\_such\_zone
- CNAME\_must\_be\_only\_record
- CNAME\_already\_on\_record
- record\_already\_exists\_not\_editable
- record\_already\_exists\_remove\_first
- internal\_error Updating zone
- internal\_error Could not load zone
- internal\_error Could not add record

### 4.4.4 File

Defined in file src/lib/dreamhost/dreamhost.lisp.

## **4.5 Dreamhost::Dns-List-Records**

### **4.5.1 Function**

Dns-List-Records names an undocumented function, with lambda list NIL.

### **4.5.2 File**

Defined in file `src/lib/dreamhost/dreamhost.lisp`.

## **4.6 Dreamhost::Dns-Remove-Record**

### **4.6.1 Function**

Dns-Remove-Record names an undocumented function, with lambda list (NAME).

### **4.6.2 File**

Defined in file `src/lib/dreamhost/dreamhost.lisp`.

## 4.7 Dreamhost::Dreamhost-Api-Error

### 4.7.1 Class

Dreamhost-Api-Error names a class, with one superclass: COMMON-LISP::ERROR (not in this manual).

### 4.7.2 Slots

Class Dreamhost-Api-Error has no direct slots defined.

## 4.8 Dreamhost::Dreamhost-Api-Error-With-Details

### 4.8.1 Class

Dreamhost-Api-Error-With-Details names a class, with one superclass: Section 4.7 [DREAMHOST DREAMHOST-API-ERROR], page 38.

A Dreamhost API error with associated details string

### 4.8.2 Slots

Class Dreamhost-Api-Error-With-Details has 1 direct slot definition:

`Details`

## 4.9 Dreamhost::Dreamhost-Api-Warning

### 4.9.1 Class

Dreamhost-Api-Warning names a class, with one superclass: COMMON-LISP::WARNING (not in this manual).

### 4.9.2 Slots

Class Dreamhost-Api-Warning has no direct slots defined.

## 4.10 Dreamhost::Dreamhost-Error-Details

### 4.10.1 Function

Dreamhost-Error-Details names an undocumented function, with lambda list (CONDITION).

## **4.11 Dreamhost::Internal-Error-Could-Not-Add-Record**

### **4.11.1 Class**

Internal-Error-Could-Not-Add-Record names a class, with one superclass: Section 4.7 [DREAMHOST DREAMHOST-API-ERROR], page 38.

### **4.11.2 Slots**

Class Internal-Error-Could-Not-Add-Record has no direct slots defined.



## 4.12 Dreamhost::Internal-Error-Could-Not-Load-Zone

### 4.12.1 Class

Internal-Error-Could-Not-Load-Zone names a class, with one superclass: Section 4.7 [DREAMHOST DREAMHOST-API-ERROR], page 38.

### 4.12.2 Slots

Class Internal-Error-Could-Not-Load-Zone has no direct slots defined.

## 4.13 Dreamhost::Internal-Error-Updating-Zone

### 4.13.1 Class

Internal-Error-Updating-Zone names a class, with one superclass: Section 4.7 [DREAM-HOST DREAMHOST-API-ERROR], page 38.

### 4.13.2 Slots

Class Internal-Error-Updating-Zone has no direct slots defined.

## 4.14 Dreamhost::Invalid-Record

### 4.14.1 Class

Invalid-Record names a class, with one superclass: Section 4.8 [DREAMHOST DREAMHOST-API-ERROR-WITH-DETAILS], page 39.

### 4.14.2 Slots

Class Invalid-Record has 1 direct slot definition:

**Details**

## 4.15 Dreamhost::Invalid-Type

### 4.15.1 Class

Invalid-Type names a class, with one superclass: Section 4.8 [DREAMHOST DREAMHOST-API-ERROR-WITH-DETAILS], page 39.

### 4.15.2 Slots

Class Invalid-Type has 1 direct slot definition:

**Details**

## 4.16 Dreamhost::Invalid-Value

### 4.16.1 Class

Invalid-Value names a class, with one superclass: Section 4.8 [DREAMHOST DREAMHOST-API-ERROR-WITH-DETAILS], page 39.

### 4.16.2 Slots

Class Invalid-Value has 1 direct slot definition:

**Details**

## **4.17 Dreamhost::No-Record**

### **4.17.1 Class**

No-Record names a class, with one superclass: Section 4.7 [DREAMHOST DREAMHOST-API-ERROR], page 38.

### **4.17.2 Slots**

Class No-Record has no direct slots defined.

## 4.18 Dreamhost::No-Such-Zone

### 4.18.1 Class

No-Such-Zone names a class, with one superclass: Section 4.7 [DREAMHOST DREAMHOST-API-ERROR], page 38.

### 4.18.2 Slots

Class No-Such-Zone has no direct slots defined.

## **4.19 Dreamhost::No-Type**

### **4.19.1 Class**

No-Type names a class, with one superclass: Section 4.7 [DREAMHOST DREAMHOST-API-ERROR], page 38.

### **4.19.2 Slots**

Class No-Type has no direct slots defined.



## 4.20 Dreamhost::No-Value

### 4.20.1 Class

No-Value names a class, with one superclass: Section 4.7 [DREAMHOST DREAMHOST-API-ERROR], page 38.

### 4.20.2 Slots

Class No-Value has no direct slots defined.

## **4.21 Dreamhost::Record-Already-Exists-Not-Editable**

### **4.21.1 Class**

Record-Already-Exists-Not-Editable names a class, with one superclass: Section 4.7 [DREAMHOST DREAMHOST-API-ERROR], page 38.

### **4.21.2 Slots**

Class Record-Already-Exists-Not-Editable has no direct slots defined.

## **4.22 Dreamhost::Record-Already-Exists-Remove-First**

### **4.22.1 Class**

Record-Already-Exists-Remove-First names a class, with one superclass: Section 4.7 [DREAMHOST DREAMHOST-API-ERROR], page 38.

### **4.22.2 Slots**

Class Record-Already-Exists-Remove-First has no direct slots defined.

## 4.23 Dreamhost::Register-Dns-Name

### 4.23.1 Function

Register-Dns-Name names an undocumented function, with lambda list (NAME IPV4-ADDRESS).

### 4.23.2 File

Defined in file `src/lib/dreamhost/dreamhost.lisp`.

## 4.24 Dreamhost::Validate-Dns-Value

### 4.24.1 Function

Validate-Dns-Value names a function, with lambda list (TYPE VALUE):

Returns VALUE in string form valid for a DNS record of type TYPE

### 4.24.2 File

Defined in file src/lib/dreamhost/dreamhost.lisp.



## 5 Package Rollbar

## 5.1 Rollbar::**\*Access-Token\***

### 5.1.1 Variable

\*Access-Token\* names a variable:

The Rollbar access-token, created through their Web UI at:

`https://rollbar.com/{team}/{project}/settings/access_tokens/`

eg:

`https://rollbar.com/CIWTA/Tootsville/settings/access_tokens/`

Its value is NIL



## 5.2 Rollbar::**\*Code-Version\***

### 5.2.1 Variable

**\*Code-Version\*** names a variable:

The version of your source code.

Can be anything, but a Git Hash is valid, as well as a software version.

Its value is NIL

## 5.3 Rollbar::**\*Environment\***

### 5.3.1 Variable

**\*Environment\*** names a variable:

The runtime environment (cluster or situational group) to report as.

Typically “development” or “production,” but more interesting labels are allowed. Groups will be automatically created by Rollbar when you report to them; no need to pre-configure anything.

Its value is `"unknown"`

## 5.4 Rollbar::**\*Framework\***

### 5.4.1 Variable

**\*Framework\*** names a variable:

Any software framework which you wish to identify as; by default, reports the name of your Lisp implementation (from LISP-IMPLEMENTATION-TYPE (see the Common Lisp HyperSpec), ie, SBCL)

Its value is "SBCL"

## 5.5 Rollbar::**\*Person-Hook\***

### 5.5.1 Variable

**\*Person-Hook\*** names a variable:

To add “person” information to a Rollbar message, create a function which examines its dynamic environment and returns a plist of the form:

```
'(:|person| (:|uid| User-UI :|username| "User name" :|email|
"user@example.com"))
```

Its value is NIL

## 5.6 Rollbar::**\*Server\***

### 5.6.1 Variable

**\*Server\*** names a variable:

The server (machine) name to report as; defaults to MACHINE-INSTANCE (see the Common Lisp HyperSpec) (which is typically the hostname)

Its value is `"Inktomi"`

## 5.7 Rollbar::**\*Valid-Notifier-Levels\***

### 5.7.1 Variable

**\*Valid-Notifier-Levels\*** names a variable:

The levels which Rollbar accepts

Its value is of type CONS

## 5.8 Rollbar::Context-Forms

### 5.8.1 Variable

+Context-Forms+ names a variable:

How many forms' worth of context should be reported?

Rollbar seems to insist upon 4.

Its value is 4 (#x4)

## 5.9 Rollbar::Backtrace-Frame-To-Plist

### 5.9.1 Function

Backtrace-Frame-To-Plist names a function, with lambda list (FRAME):

Convert FRAME into a plist of the sort Rollbar likes

### 5.9.2 File

Defined in file src/lib/rollbar/rollbar.lisp.



## 5.10 Rollbar::Chain-Debugger-Hook

### 5.10.1 Function

Chain-Debugger-Hook names a function, with lambda list NIL:

Create a function that calls 'DEBUGGER-HOOK'.

The present value of \*DEBUGGER-HOOK\* (see the Common Lisp HyperSpec) is closed over by that function, and will be called after calling 'DEBUGGER-HOOK'.

### 5.10.2 File

Defined in file `src/lib/rollbar/rollbar.lisp`.

## 5.11 Rollbar::Classify-Error-Level

### 5.11.1 Function

Classify-Error-Level names a function, with lambda list (CONDITION):

Given CONDITION, return the Rollbar level for it.

Methods can specialize on condition types to return specific levels, but the defaults should be fairly sane for most users.

Note that SERIOUS-CONDITION (see the Common Lisp HyperSpec) maps to “error,” while ERROR (see the Common Lisp HyperSpec) maps to “critical,” to more closely match Rollbar’s system.

### 5.11.2 File

Defined in file src/lib/rollbar/rollbar.lisp.

## **5.12 Rollbar::Condition-Telemetry**

### **5.12.1 Function**

Condition-Telemetry names an undocumented function, with lambda list (CONDITION).

### **5.12.2 File**

Defined in file `src/lib/rollbar/rollbar.lisp`.

## 5.13 Rollbar::Configure

### 5.13.1 Function

Configure names a function, with lambda list (&KEY (ACCESS-TOKEN NIL ACCESS-TOKEN-PRESENT-P) (ENVIRONMENT NIL ENVIRONMENT-PRESENT-P) (CODE-VERSION NIL CODE-VERSION-PRESENT-P) (FRAMEWORK NIL FRAMEWORK-PRESENT-P) (SERVER NIL SERVER-PRESENT-P)):

Sets Rollbar configuration persistently (dynamically).

Typically only invoked once at startup.

### 5.13.2 File

Defined in file `src/lib/rollbar/rollbar.lisp`.

## **5.14 Rollbar::Constituent-Char-P**

### **5.14.1 Function**

Constituent-Char-P names an undocumented function, with lambda list (CHAR).

### **5.14.2 File**

Defined in file `src/lib/rollbar/rollbar.lisp`.

## 5.15 Rollbar::Critical!

### 5.15.1 Function

Critical! names a function, with lambda list (MESSAGE\* &KEY CONDITION):

Report a condition to Rollbar with level critical.

Calls 'NOTIFY' like (NOTIFY "critical" MESSAGE ...).

The ! in the name is so that ROLLBAR:ERROR! does not shadow CL:ERROR, and so that all levels share the same orthography.

### 5.15.2 File

Defined in file dumper-2SKVI5f7.lisp.

## 5.16 Rollbar::Debug!

### 5.16.1 Function

Debug! names a function, with lambda list (MESSAGE\* &KEY CONDITION):

Report a condition to Rollbar with level debug.

Calls 'NOTIFY' like (NOTIFY "debug" MESSAGE . . .).

The ! in the name is so that ROLLBAR:ERROR! does not shadow CL:ERROR, and so that all levels share the same orthography.

### 5.16.2 File

Defined in file dumper-2SKVI5f7.lisp.

## 5.17 Rollbar::Debugger-Hook

### 5.17.1 Function

Debugger-Hook names a function, with lambda list (CONDITION &OPTIONAL HOOK):

Take the CONDITION reported to the debugger and relay it to Rollbar.

This is usually activated through ‘WITH-ROLLBAR-FOR-DEBUGGER’.

### 5.17.2 File

Defined in file src/lib/rollbar/rollbar.lisp.



## 5.18 Rollbar::Error!

### 5.18.1 Function

Error! names a function, with lambda list (MESSAGE\* &KEY CONDITION):

Report a condition to Rollbar with level error.

Calls 'NOTIFY' like (NOTIFY "error" MESSAGE . . .).

The ! in the name is so that ROLLBAR:ERROR! does not shadow CL:ERROR, and so that all levels share the same orthography.

### 5.18.2 File

Defined in file dumper-2SKVI5f7.lisp.

## 5.19 Rollbar::Escaped

### 5.19.1 Function

Escaped names a function, with lambda list (STRING ESCAPE-CHAR ESCAPED-CHARS):

Escape characters within the string, usually by \

### 5.19.2 File

Defined in file src/lib/rollbar/rollbar.lisp.

## 5.20 Rollbar::Find-Appropriate-Backtrace

### 5.20.1 Function

Find-Appropriate-Backtrace names a function, with lambda list NIL:

Finds a backtrace without too much “noise.”

Attempts to eliminate “uninteresting” frames from the trace, and formats it in a form that Rollbar likes.

### 5.20.2 File

Defined in file `src/lib/rollbar/rollbar.lisp`.

## **5.21 Rollbar::Format-Symbol-Name-Carefully**

### **5.21.1 Function**

Format-Symbol-Name-Carefully names a function, with lambda list (SYMBOL):

Carefully format the symbol-name of SYMBOL

### **5.21.2 File**

Defined in file src/lib/rollbar/rollbar.lisp.

## 5.22 Rollbar::Gather-Source-Info

### 5.22.1 Function

Gather-Source-Info names a function, with lambda list (FILENAME TOP-LEVEL-FORM FORM-NUMBER):

Get source code information for a frame in a backtrace

### 5.22.2 File

Defined in file src/lib/rollbar/rollbar.lisp.

## 5.23 Rollbar::Http-Error

### 5.23.1 Class

Http-Error names a class, with one superclass: COMMON-LISP::ERROR (not in this manual).

### 5.23.2 Slots

Class Http-Error has 5 direct slot definitions:

Status

Status-Text

Wanted-Uri

Got-Uri

Headers

## **5.24 Rollbar::Http-Error-Got-Uri**

### **5.24.1 Function**

Http-Error-Got-Uri names an undocumented function, with lambda list (CONDITION).

## 5.25 Rollbar::Http-Error-Headers

### 5.25.1 Function

Http-Error-Headers names an undocumented function, with lambda list (CONDITION).



## 5.26 Rollbar::Http-Error-Status

### 5.26.1 Function

Http-Error-Status names an undocumented function, with lambda list (CONDITION).

## 5.27 Rollbar::Http-Error-Status-Text

### 5.27.1 Function

Http-Error-Status-Text names an undocumented function, with lambda list (CONDITION).

## 5.28 Rollbar::Http-Error-Wanted-Uri

### 5.28.1 Function

Http-Error-Wanted-Uri names an undocumented function, with lambda list (CONDITION).

## 5.29 Rollbar::Http-Successful-Request

### 5.29.1 Function

Http-Successful-Request names an undocumented function, with lambda list (URI &REST KEYS &KEY &ALLOW-OTHER-KEYS).

### 5.29.2 File

Defined in file `src/lib/rollbar/rollbar.lisp`.

## 5.30 Rollbar::Info!

### 5.30.1 Function

Info! names a function, with lambda list (MESSAGE\* &KEY CONDITION):

Report a condition to Rollbar with level info.

Calls 'NOTIFY' like (NOTIFY "info" MESSAGE . . .).

The ! in the name is so that ROLLBAR:ERROR! does not shadow CL:ERROR, and so that all levels share the same orthography.

### 5.30.2 File

Defined in file dumper-2SKVI5f7.lisp.

## **5.31 Rollbar::Level-Is-Valid-P**

### **5.31.1 Function**

Level-Is-Valid-P names a function, with lambda list (LEVEL):

Determines whether LEVEL is a valid level indicator for Rollbar.

### **5.31.2 File**

Defined in file src/lib/rollbar/rollbar.lisp.

## 5.32 Rollbar::Make-Level-Notifier

### 5.32.1 Function

Make-Level-Notifier names an undocumented function, with lambda list (LEVEL).

### 5.32.2 File

Defined in file `src/lib/rollbar/rollbar.lisp`.

## 5.33 Rollbar::Notify

### 5.33.1 Function

Notify names a function, with lambda list (LEVEL MESSAGE\* &KEY CONDITION):

Sends a notification to Rollbar of level LEVEL with message MESSAGE\*.

If CONDITION is given, useful information is extracted therefrom (eg, backtrace).

Without CONDITION, the backtrace will be from the current (caller) context.

If unable to reach Rollbar, a SIGNAL of type CAN-NOT-REPORT will be raised, which you can choose to CATCH or ignore.

A log entry will also be printed to \*TRACE-OUTPUT\* for levels “debug” or “info,” and to \*ERROR-OUTPUT\* for other levels. (See ‘OUTPUT-FOR-LEVEL’)

### 5.33.2 File

Defined in file src/lib/rollbar/rollbar.lisp.



## 5.34 Rollbar::Output-For-Level

### 5.34.1 Function

Output-For-Level names a function, with lambda list (LEVEL):

Returns a stream for logging messages of level LEVEL.

For “info” or “debug,” returns \*TRACE-OUTPUT\*; otherwise \*ERROR-OUTPUT\*.

### 5.34.2 File

Defined in file src/lib/rollbar/rollbar.lisp.

## 5.35 Rollbar::Package-Name-Can-Be-Unquoted-P

### 5.35.1 Function

Package-Name-Can-Be-Unquoted-P names a function, with lambda list (PACKAGE-NAME):

Decide whether a package name symbol can be printed without quoting

### 5.35.2 File

Defined in file src/lib/rollbar/rollbar.lisp.

## 5.36 Rollbar::Pretty-Function-Name

### 5.36.1 Function

Pretty-Function-Name names a function, with lambda list (FUNCTION):

Pretty-print the name (and type information) of FUNCTION

### 5.36.2 File

Defined in file src/lib/rollbar/rollbar.lisp.

## 5.37 Rollbar::Pretty-Symbol-Name

### 5.37.1 Function

Pretty-Symbol-Name names a function, with lambda list (SYMBOL):

Format the symbol-name of SYMBOL nicely for the Rollbar report

### 5.37.2 File

Defined in file src/lib/rollbar/rollbar.lisp.

## **5.38 Rollbar::Quoted**

### **5.38.1 Function**

Quoted names a function, with lambda list (STRING):

Return a quoted version of String

### **5.38.2 File**

Defined in file src/lib/rollbar/rollbar.lisp.

## 5.39 Rollbar::Redact-Directory

### 5.39.1 Function

Redact-Directory names a function, with lambda list (DIRECTORY):

Redact uninteresting parts of a directory pathname

### 5.39.2 File

Defined in file src/lib/rollbar/rollbar.lisp.

## **5.40 Rollbar::Report-Server-Info**

### **5.40.1 Function**

Report-Server-Info names a function, with lambda list NIL:

Generate the server-info Plist for the error report

### **5.40.2 File**

Defined in file src/lib/rollbar/rollbar.lisp.

## **5.41 Rollbar::Report-Telemetry**

### **5.41.1 Function**

Report-Telemetry names a function, with lambda list (LEVEL):

Generates some general information for the error report

### **5.41.2 File**

Defined in file `src/lib/rollbar/rollbar.lisp`.



## **5.42 Rollbar::Request-Telemetry**

### **5.42.1 Function**

Request-Telemetry names an undocumented function, with lambda list NIL.

### **5.42.2 File**

Defined in file `src/lib/rollbar/rollbar.lisp`.

## 5.43 Rollbar::Rollbar-Notify-Deployment

### 5.43.1 Function

Rollbar-Notify-Deployment names an undocumented function, with lambda list (&KEY USER REVISION ENVIRONMENT).

### 5.43.2 File

Defined in file src/lib/rollbar/rollbar.lisp.

## 5.44 Rollbar::Sanitize-File-Name

### 5.44.1 Function

Sanitize-File-Name names an undocumented function, with lambda list (PATHNAME).

### 5.44.2 File

Defined in file `src/lib/rollbar/rollbar.lisp`.

## 5.45 Rollbar::Send-Rollbar-Notification

### 5.45.1 Function

Send-Rollbar-Notification names a function, with lambda list (LEVEL MESSAGE BACKTRACE &KEY CONDITION):

Send a notification to Rollbar.

### 5.45.2 File

Defined in file src/lib/rollbar/rollbar.lisp.

## 5.46 Rollbar::Symbol-Is-Exported-P

### 5.46.1 Function

Symbol-Is-Exported-P names a function, with lambda list (SYMBOL):

Discover whether SYMBOL is exported from its package

### 5.46.2 File

Defined in file src/lib/rollbar/rollbar.lisp.

## 5.47 Rollbar::Symbol-Name-Can-Be-Unquoted-P

### 5.47.1 Function

Symbol-Name-Can-Be-Unquoted-P names a function, with lambda list (SYMBOL):

Decide whether a symbol name can be printed without quoting

### 5.47.2 File

Defined in file src/lib/rollbar/rollbar.lisp.

## 5.48 Rollbar::Warning!

### 5.48.1 Function

Warning! names a function, with lambda list (MESSAGE\* &KEY CONDITION):

Report a condition to Rollbar with level warning.

Calls 'NOTIFY' like (NOTIFY "warning" MESSAGE ...).

The ! in the name is so that ROLLBAR:ERROR! does not shadow CL:ERROR, and so that all levels share the same orthography.

### 5.48.2 File

Defined in file dumper-2SKVI5f7.lisp.

## 5.49 Rollbar::With-Configuration

### 5.49.1 Macro

With-Configuration names a macro, with lambda list ((&REST KEYS &KEY ACCESS-TOKEN ENVIRONMENT CODE-VERSION FRAMEWORK SERVER) &BODY BODY):

Executes BODY with Rollbar variables bound to the values given (if any).

Unmentioned keys are left unaltered.

### 5.49.2 File

Defined in file src/lib/rollbar/rollbar.lisp.



## 5.50 Rollbar::With-Rollbar-For-Debugger

### 5.50.1 Macro

With-Rollbar-For-Debugger names a macro, with lambda list (NIL &BODY BODY):

Run BODY with `*DEBUGGER-HOOK*` (see the Common Lisp HyperSpec) bound to call Rollbar.

Any previous value of `*DEBUGGER-HOOK*` (see the Common Lisp HyperSpec) will be called after Rollbar.

### 5.50.2 File

Defined in file `src/lib/rollbar/rollbar.lisp`.



## 6 Package Thread-Pool-Taskmaster

## **6.1 Thread-Pool-Taskmaster::**\*Developmentp\*****

### **6.1.1 Variable**

**\*Developmentp\*** names an undocumented variable with the value NIL

## 6.2 Thread-Pool-Taskmaster::**\*Mulligans\***

### 6.2.1 Variable

**\*Mulligans\*** names an undocumented variable with the value 5 (`#x5`)

## **6.3 Thread-Pool-Taskmaster::+Max-Queue-Size-For-Thread-Pool+**

### **6.3.1 Variable**

+Max-Queue-Size-For-Thread-Pool+ names a variable:

What is the maximum queue size allowed for a thread pool?

Its value is 256 (#x100)

## **6.4 Thread-Pool-Taskmaster::+Single-Core-Threads+**

### **6.4.1 Variable**

+Single-Core-Threads+ names a variable:

More threads than otherwise expected on a single-core machine.

Its value is 16 (#x10)

## 6.5 Thread-Pool-Taskmaster::+Threads-Per-Core+

### 6.5.1 Variable

+Threads-Per-Core+ names a variable:

Must be an (UNSIGNED-BYTE 15) and non-zero.

Its value is 8 (#x8)



## 6.6 Thread-Pool-Taskmaster::Cores\*Threads-Per-Core

### 6.6.1 Function

Cores\*Threads-Per-Core names an undocumented function, with lambda list (&REST ARGUMENTS).

### 6.6.2 File

Defined in file quicklisp/dists/quicklisp/software/fare-memoization-20180430-git/memoization.lisp.

## **6.7 Thread-Pool-Taskmaster::Make-Thread-Name**

### **6.7.1 Function**

Make-Thread-Name names an undocumented function, with lambda list (TASKMASTER SOCKET).

### **6.7.2 File**

Defined in file `src/lib/taskmaster/thread-pool-taskmaster.lisp`.

## **6.8 Thread-Pool-Taskmaster::Name-Idle-Threads-Sequentially**

### **6.8.1 Function**

Name-Idle-Threads-Sequentially names an undocumented function, with lambda list (TASKMASTER).

### **6.8.2 File**

Defined in file `src/lib/taskmaster/thread-pool-taskmaster.lisp`.

## **6.9 Thread-Pool-Taskmaster::Named-Thread-Pool-Runner**

### **6.9.1 Macro**

Named-Thread-Pool-Runner names an undocumented macro, with lambda list ((&KEY (NAME)) &BODY BODY).

### **6.9.2 File**

Defined in file `src/lib/taskmaster/thread-pool-taskmaster.lisp`.

## **6.10 Thread-Pool-Taskmaster::Safe-Client-As-String**

### **6.10.1 Function**

Safe-Client-As-String names an undocumented function, with lambda list (SOCKET).

### **6.10.2 File**

Defined in file `src/lib/taskmaster/thread-pool-taskmaster.lisp`.

## **6.11 Thread-Pool-Taskmaster::Swank-Connected-P**

### **6.11.1 Function**

Swank-Connected-P names a function, with lambda list NIL:

Detect whether Swank is connected.

Used to determine whether to resignal errors.

### **6.11.2 File**

Defined in file `src/lib/taskmaster/thread-pool-taskmaster.lisp`.

## 6.12 Thread-Pool-Taskmaster::Taskmaster-Thread-Pool

### 6.12.1 Function

Taskmaster-Thread-Pool names an undocumented function, with lambda list (OBJECT).

### 6.12.2 SetF Function

(SETF Taskmaster-Thread-Pool) names an undocumented function, with lambda list (NEW-VALUE OBJECT).

## **6.13 Thread-Pool-Taskmaster::Taskmaster-Thread-Pool-Channel**

### **6.13.1 Function**

Taskmaster-Thread-Pool-Channel names an undocumented function, with lambda list (OBJECT).

### **6.13.2 SetF Function**

(SETF Taskmaster-Thread-Pool-Channel) names an undocumented function, with lambda list (NEW-VALUE OBJECT).



## 6.14 Thread-Pool-Taskmaster::Thread-Pool-Taskmaster

### 6.14.1 Class

Thread-Pool-Taskmaster names a class, with one superclass: HUNCHENTOOT::ONE-THREAD-PER-CONNECTION-TASKMASTER (not in this manual).

A taskmaster that uses a thread pool to dispatch incoming requests.

### 6.14.2 Slots

Class Thread-Pool-Taskmaster has no direct slots defined.

## **6.15 Thread-Pool-Taskmaster::With-Mulligan-Handlers**

### **6.15.1 Macro**

With-Mulligan-Handlers names an undocumented macro, with lambda list ((NAME MULLIGAN) &BODY BODY).

### **6.15.2 File**

Defined in file `src/lib/taskmaster/thread-pool-taskmaster.lisp`.

## **6.16 Thread-Pool-Taskmaster::With-Pool-Thread-Restarts**

### **6.16.1 Macro**

With-Pool-Thread-Restarts names an undocumented macro, with lambda list ((NAME) &BODY BODY).

### **6.16.2 File**

Defined in file `src/lib/taskmaster/thread-pool-taskmaster.lisp`.



## 7 Package Tootsville-User

## 7.1 Tootsville-User::\$

### 7.1.1 Function

\$ names a function, with lambda list (&REST WORDS):

Execute a command script.

### 7.1.2 Usage

#\$ SCRIPT-NAME

### 7.1.3 Example

#\$ do-something-funny

The script name must be a function previously defined by #SCRIPT; see Section 7.72 [TOOTSVILLE-USER SCRIPT], page 203, operator command for details.

### 7.1.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.2 Tootsville-User::\*Apropos

### 7.2.1 Function

\*Apropos names a function, with lambda list (&REST WORDS):

Runs APROPOS (see the Common Lisp HyperSpec) for a remote user.

### 7.2.2 Usage

```
#apropos EXPRESSION
```

### 7.2.3 Example

```
#apropos apropos
```

### 7.2.4 File

Defined in file src/infinity/modern-ops.lisp.

## 7.3 Tootsville-User::\*Time

### 7.3.1 Function

\*Time names a function, with lambda list (&REST WORDS):

Displays a message with the current server time.

This is a convenience function to ask “what time is it,” as opposed to the verb “time this to see how long it takes” like TIME (see the Common Lisp HyperSpec).

### 7.3.2 Usage

```
#time
```

### 7.3.3 Example

```
#time
```

### 7.3.4 Example Reply

```
Now it is 2021-01-26T00:35:11.341489Z (Universal: 3,820,610,111; Unix:  
1,611,621,311). In Chærogryllum, it is 0:35:11 on Blanksday, the  
eleventh of Procavia, 153
```

### 7.3.5 Changes from 1.2 to 2.0

The output format has changed. The old version only displayed the Unix time in seconds, without commas; the decoded date and time, Universal time code, and Chærogryllum date and time are new.

### 7.3.6 File

Defined in file src/infinity/legacy-ops.lisp.



## 7.4 Tootsville-User::\*Warn

### 7.4.1 Function

\*Warn names a function, with lambda list (&REST WORDS):

Warn a user about breaking a rule.

Warns a user (anonymously) about the Tootsville rules. The warning messages are pre-determined canned messages accessed via short mnemonic names.

To obtain the list of mnemonics, type `#warn #list`.

### 7.4.2 Usage

```
#warn #list  
#warn REASONCODE LOGIN
```

### 7.4.3 Examples

```
#warn #list  
#warn BULLY Pil
```

### 7.4.4 Reason Codes

See Section 7.50 [TOOTSVILLE-USER KICK], page 177, for the current list

### 7.4.5 Changes from 1.2 to 2.0

This command's reason codes have changed from 1.2 to 2.0 completely. The new list is kept under Section 7.50 [TOOTSVILLE-USER KICK], page 177.

`#warn #list` is a new command.

### 7.4.6 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.5 Tootsville-User::Addevent

### 7.5.1 Function

Addevent names a function, with lambda list (&REST WORDS):

Add a GameEvent to a Zone

UNIMPLEMENTED

### 7.5.2 Usage

#addevent [EVENTNAME]

### 7.5.3 Examples

```
#addevent LaserTagGame
```

```
#addevent PropsWeather
```

```
#addevent ShaddowFalls
```

```
#addevent Tootlympics
```

This is not currently implemented.

### 7.5.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.6 Tootsville-User::Agent

### 7.6.1 Function

Agent names a function, with lambda list (&REST WORDS):

Set the clothing and colors of a robot to match the invoking user.

### 7.6.2 Usage

```
#agent robot-name
```

### 7.6.3 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.7 Tootsville-User::Askme

### 7.7.1 Function

Askme names a function, with lambda list (&REST WORDS):

Used to test the question-and-answer subsystem.

### 7.7.2 Usage

```
#askme
```

### 7.7.3 200 OK

Returns a fixed JSON sequence that prompts the user to answer a meaningless question.

```
{ title: "Title Here",
  label: "example",
  label_en_US: "example",
  attachUser: (the user name of the invoking user),
  id: "example/2134$p=?/x'<>'\\\",:/blah",
  msg: "Because it's really important to me that you are able to hear this question and give
  replies:
  { si: { label: "Yes",
          type: "aff",
          label_en_US: "YES" },
    no: { label: "No",
          type: "neg",
          label_en_US: "NO" },
    maybe: { label: "Maybe. I'm not really sure. This one is mostly just in here to be a re
              type: "neu",
              label_en_US: "MEBBE" } } }
```

### 7.7.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.8 Tootsville-User::At

### 7.8.1 Function

At names a function, with lambda list (&REST WORDS):

Issue an operator command on a particular server instance.

UNIMPLEMENTED. Remote code execution is not possible. Only works if SERVER is MACHINE-INSTANCE (see the Common Lisp HyperSpec), i.e. the local machine, which is identical to not using this command at all.

### 7.8.2 Usage

```
#at SERVER #OTHER-COMMAND OTHER-PARAMS
```

```
#at #each #OTHER-COMMAND OTHER-PARAMS
```

### 7.8.3 Examples

```
#at game1.test.tootsville.net #ws-stats
```

```
#at #each #git-pull
```

For a list of servers, see Section 8.1120 [TOOTSVILLE SERVER-LIST], page 1405.

To issue a command on every server, send `#at #each #OTHER-COMMAND`.

### 7.8.4 File

Defined in file `src/infinity/modern-ops.lisp`.

## 7.9 Tootsville-User::Ban

### 7.9.1 Function

Ban names a function, with lambda list (&REST WORDS):

Ban a user persistently (permanently) from the game.

### 7.9.2 Usage

```
#ban [REASONCODE] [LOGIN]
#ban #list
```

### 7.9.3 Examples

```
#ban BULLY pil
#ban #list
```

`#ban #list` is identical to `#kick #list`, for convenience.

The same reason codes and syntax are used for `#ban` as for `#kick`, so refer to Section 7.50 [TOOTSVILLE-USER KICK], page 177, for reason codes.

Unlike a `#kick`, a `#ban` remains in effect persistently — permanently, unless an operator reverses it.

### 7.9.4 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.10 Tootsville-User::Banhammer

### 7.10.1 Function

Banhammer names a function, with lambda list (&REST WORDS):

Ban an IP address from connecting.

Bans can be listed using `#banhammer #list`

Bans can be lifted using `#banhammer #-ip IP-ADDRESS` (or hostname)

A ban can be placed with `#banhammer #+ip IP-ADDRESS` or `#banhammer #+ip HOSTNAME` or `#banhammer #user USERNAME`. In the latter case, the user's connected IP address is used. This is expected to be the most common usage.

### 7.10.2 Usage

```
#banhammer #list
```

```
#banhammer #user NICKNAME
```

```
#banhammer #+ip ADDRESS
```

```
#banhammer #-ip ADDRESS
```

### 7.10.3 Parameters

The first word is a subcommand; one of `#+ip`, `#-ip`, `#user`, or `#list`. For `#+ip`, `#-ip`, or `#user`, an additional parameter is needed.

### 7.10.4 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.11 Tootsville-User::Beam

### 7.11.1 Function

Beam names a function, with lambda list (&REST WORDS):

Beam yourself to a different location.

### 7.11.2 Usage

```
#beam LATITUDE LONGITUDE [ALTITUDE]
```

Altitude is optional.

### 7.11.3 Changes from 1.2 to 2.0

In Romance 1, this command took a room moniker as its sole parameter; since rooms as such no longer exist, we use latitude and longitude now.

### 7.11.4 File

Defined in file src/infinity/legacy-ops.lisp.



## 7.12 Tootsville-User::Census

### 7.12.1 Function

Census names a function, with lambda list (&REST WORDS):

Load a number of users.

Simply reference a range of users, for testing purposes. Takes an optional low and high point, or runs 0..250000. (250,000) This will load their Toots, and seriously strain the caché and database subsystems.

Afterwards, due to cache flooding, database accesses may be slower than usual until things balance out to a more normal workload.

Since this is designed to stress the servers, it can be called only by God (Pil).

### 7.12.2 Usage

```
#census
```

```
#census [START]
```

```
#census [START] [COUNT]
```

### 7.12.3 Examples

```
#census
```

Stress-load the first 250,000 Toots

```
#census 20000
```

Stress-load 250,000 Toots starting with offset 20,000.

```
#census 1000 100
```

Stress-load 100 Toots starting with offset 1,000.

### 7.12.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.13 Tootsville-User::Clearbadge

### 7.13.1 Function

Clearbadge names a function, with lambda list (&REST WORDS):

Clear a badge off the map.

UNIMPLEMENTED: This is not yet implemented for Tootsville V.

### 7.13.2 Usage

```
#clearbadge [NICKNAME] [SPOT]
```

```
#clearbadge #me [SPOT]
```

```
#clearbadge #me #here
```

```
#clearbadge [NICKNAME] #here
```

```
#clearbadge #me #all
```

```
#clearbadge [NICKNAME] #all
```

### 7.13.3 Examples

```
#clearbadge snowcone tootSquare
```

```
#clearbadge snowcone #all
```

```
#clearbadge snowcone #here
```

```
#clearbadge #me #all
```

```
#clearbadge #me #here
```

### 7.13.4 Badges

See Section 7.75 [TOOTSVILLE-USER SETBADGE], page 206, for a discussion of the map badges system.

### 7.13.5 Spots

See Section 7.82 [TOOTSVILLE-USER SPAWNROOM], page 213, to mark a spot with a moniker, so that it can be passed to #clearbadge and other “spot-based” commands.

### 7.13.6 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.14 Tootsville-User::Clearcache

### 7.14.1 Function

Clearcache names a function, with lambda list (&REST WORDS):

Forcibly clear all caches (MemCacheD)

Flush all contents of the MemCacheD server. This may negatively impact the system's performance.

### 7.14.2 Usage

```
#clearcache
```

### 7.14.3 Example

```
#clearcache
```

### 7.14.4 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.15 Tootsville-User::Clearevent

### 7.15.1 Function

Clearevent names a function, with lambda list (&REST WORDS):

```
Clear a GameEvent  
UNIMPLEMENTED
```

### 7.15.2 Usage

```
#clearevent [EVENTNAME] [UNIQUE ID]
```

### 7.15.3 Examples

```
#clearevent LaserTagGame 142  
#clearevent PropsWeather 120  
#clearevent ShaddowFalls 928  
#clearevent Tootlympics 1294
```

### 7.15.4 Changes from 1.2 to 2.0

The names of game events have changed format.

The unique ID parameter is now required.

### 7.15.5 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.16 Tootsville-User::Clearvar

### 7.16.1 Function

Clearvar names a function, with lambda list (&REST WORDS):

Clear a room variable. (no longer supported)

Room variables can no longer be cleared. This command is no longer useful.

### 7.16.2 Usage

```
#clearvar @[ROOM] [VARIABLE] [VALUE]
```

```
#clearvar [VARIABLE] [VALUE]
```

### 7.16.3 Examples

```
#clearvar @tootsSquareWest anim~ropes 2
```

```
#clearvar anim~ropes 2
```

### 7.16.4 410 Gone

This command was rendered inoperable in 2.0.

### 7.16.5 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.17 Tootsville-User::Clonerroom

### 7.17.1 Function

Clonerroom names a function, with lambda list (&REST WORDS):

Clone a room. (no longer supported)

This is no longer supported in Tootsville V.

### 7.17.2 Usage

```
#clonerroom NEW-MONIKER
```

```
#clonerroom OLD-MONIKER NEW-MONIKER
```

### 7.17.3 Legacy Operator Command

This command existed in Romance 1.2, but is no longer effective.

### 7.17.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.18 Tootsville-User::Createroom

### 7.18.1 Function

Createroom names a function, with lambda list (&REST WORDS):

Create a new room.

This is a synonym for Section 7.82 [TOOTSVILLE-USER SPAWNROOM], page 213, now.

### 7.18.2 Usage

```
#createroom NEW-MONIKER
```

### 7.18.3 Example

```
#createroom JACKS-HOUSE  
#spawnroom JACKS-HOUSE
```

### 7.18.4 Legacy Operator Command

This command existed in Romance 1.2 and was different than Section 7.82 [TOOTSVILLE-USER SPAWNROOM], page 213, but now they are synonymous.

### 7.18.5 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.19 Tootsville-User::Dbcpinfo

### 7.19.1 Function

Dbcpinfo names a function, with lambda list (&REST WORDS):

Get information from the DBI (database) layer.

Earlier versions of Romance were Java-based, using the DBCP layer, thus the name.

### 7.19.2 Usage

`#dbcpinfo`

Examples:

`#dbcpinfo`

This identifies the name of the database being used, the DBI driver type, and the active connection or connection pool.

### 7.19.3 File

Defined in file `src/infinity/legacy-ops.lisp`.



## 7.20 Tootsville-User::Doc

### 7.20.1 Function

Doc names a function, with lambda list (&REST WORDS):

Obtain documentation string in raw form about a symbol.

### 7.20.2 Usage

```
#doc [PACKAGE] SYMBOL [TYPE]
```

TYPE can be VARIABLE, FUNCTION, STRUCTURE, TYPE, SETF, or T. If not supplied, defaults to FUNCTION.

PACKAGE is optional and defaults to TOOTSVILLE-USER.

### 7.20.3 Examples

```
#doc cdr
#doc doc function
#doc Tootsville ws-stats
#doc Tootsville ws-bandwidth-by-source function
```

This is based upon DOCUMENTATION (see the Common Lisp HyperSpec), qv.

### 7.20.4 File

Defined in file src/infinity/modern-ops.lisp.

## 7.21 Tootsville-User::Doodle

### 7.21.1 Function

Doodle names a function, with lambda list (&REST WORDS):

Change the colors of a Toot.

### 7.21.2 Usage

```
#doodle WHO ( #base | #pad | #pattern ) NEW-COLOR
```

### 7.21.3 Examples

```
#doodle catvllle base pink
#doodle catvllle pattern black
```

The Toot's color will immediately be changed and be advertised to any interested listener.

NEW-COLOR can be in any format understood by Section 8.945 [TOOTSVILLE PARSE-COLOR24], page 1230, qv.

### 7.21.4 File

Defined in file src/infinity/modern-ops.lisp.

## 7.22 Tootsville-User::Doodle-Pattern

### 7.22.1 Function

Doodle-Pattern names a function, with lambda list (&REST WORDS):

Change the pattern of a Toot.

### 7.22.2 Usage

```
#doodle-pattern WHO NEW-PATTERN
```

### 7.22.3 Example

```
#doodle-pattern catvllle hearts
```

As a special case, "Polka Dots" should be passed as POLKA-DOTS (with an hyphen), as well as any other pattern names with spaces (such as "Maple Leaf").

### 7.22.4 File

Defined in file src/infinity/modern-ops.lisp.

## 7.23 Tootsville-User::Dress

### 7.23.1 Function

Dress names a function, with lambda list (&REST WORDS):

Force a character to wear a specific clothing item.

UNIMPLEMENTED

### 7.23.2 Usage

```
#dress [LOGIN] [ITEM]
```

```
#dress [LOGIN] [ITEM] [COLOUR]
```

### 7.23.3 Examples

```
#dress flappyperry 1337
```

### 7.23.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.24 Tootsville-User::Drop

### 7.24.1 Function

Drop names a function, with lambda list (&REST WORDS):

Drop an item

UNIMPLEMENTED

Find an item in your inventory based upon the item ID # and drop it (to the world).

Usage: #drop ITEM-TEMPLATE-ID

### 7.24.2 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.25 Tootsville-User::Dropkick

### 7.25.1 Function

Dropkick names a function, with lambda list (&REST WORDS):

Silently disconnect a user

UNIMPLEMENTED

Silently remove the named user from the game by disconnection. Must have staff level 4 (DESIGNER) to use this command.

### 7.25.2 Usage

```
#dropkick [LOGIN]
```

### 7.25.3 Example

```
#dropkick flappyperry
```

### 7.25.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.26 Tootsville-User::Dumpthreads

### 7.26.1 Function

Dumpthreads names a function, with lambda list (&REST WORDS):

Dump debugging information including all running threads to the server logs.

### 7.26.2 Usage

```
#dumpthreads  
,dumpthreads
```

### 7.26.3 Example

```
#dumpthreads  
,dumpthreads
```

Note that this can be invoked as `,dumpthreads` by a non-operator user as well.

### 7.26.4 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.27 Tootsville-User::Enablepathfinder

### 7.27.1 Function

Enablepathfinder names a function, with lambda list (&REST WORDS):

Temporary test routine for testing pathfinders on users

UNIMPLEMENTED

### 7.27.2 Usage

`#enablepathfinder (true|false)`

### 7.27.3 Examples

`#enablepathfinder true`

`#enablepathfinder false`

### 7.27.4 File

Defined in file `src/infinity/legacy-ops.lisp`.



## 7.28 Tootsville-User::Evacuate

### 7.28.1 Function

Evacuate names a function, with lambda list (&REST WORDS):

Evacuate all users from the current Zone to another Zone.

UNIMPLEMENTED

Evacuate all users from your current server into another server. Will error if the server specified does not exist in the cluster.

### 7.28.2 Usage

```
#evacuate [SERVER]
```

### 7.28.3 Example

```
#evacuate game2
```

### 7.28.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.29 Tootsville-User::Filter

### 7.29.1 Function

Filter names a function, with lambda list (&REST WORDS):

Test censorship rules against words or phrases

### 7.29.2 Usage

```
#filter EXPRESSION
```

```
#filter #all EXPRESSION
```

```
#filter #child EXPRESSION
```

See Section 8.178 [TOOTSVILLE CASSANDRA-FILTER], page 432, et al.

### 7.29.3 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.30 Tootsville-User::Finger

### 7.30.1 Function

Finger names a function, with lambda list (&REST WORDS):

Finger a user account.

Return interesting details in an administrative message.

### 7.30.2 Usage

```
#finger TOOT
```

### 7.30.3 Examples

```
#finger mouser
```

```
#finger shade
```

### 7.30.4 Changes from 1.2 to 2.0

The format of the response has changed slightly, but is similar.

### 7.30.5 Response

Mouser is a Toot with base color red, pad color black, and pattern black spots. This is an adult's account. (sensitive player) (patron) The user has 2,130 peanuts, 100 fairy dust, and was last active @2021-01-26T04:02:55.600079Z (Earth time; two minutes ago; Blanksday, the eleventh of Procavia, 153) The player owning Mouser is John Doe (jdoe@gmail.com). Toot: 5112AE4B-0F8D-4823-AFD7-EC4119001D04, player: AC14ABCF-518D-4DC5-B783-3A4DFE4838B2

### 7.30.6 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.31 Tootsville-User::Flush

### 7.31.1 Function

Flush names a function, with lambda list (&REST WORDS):

Historically, this flushed the database write caché.

This command is not currently implemented. It is a no-op.

### 7.31.2 Usage

`#flush`

### 7.31.3 Changes from 1.2 to 2.0

This command is not effective in Romance II. In Romance 1, it was used to flush the database write caché, which at times could be several minutes behind the database's on-disk version. This should no longer be a concern in 2020 and beyond, but the command is retained as a no-op. It might be re-activated in future as needed.

### 7.31.4 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.32 Tootsville-User::Game

### 7.32.1 Function

Game names a function, with lambda list (&REST WORDS):

Send a command to the operator command interpreter for a running game.

UNIMPLEMENTED

Send a command into the operator command interpreter for a running game (if that game provides one)

### 7.32.2 Usage

```
#game gameClass (strings...)
```

### 7.32.3 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.33 Tootsville-User::Gc

### 7.33.1 Function

Gc names a function, with lambda list (&REST WORDS):

Perform immediate garbage collection.

### 7.33.2 Usage

```
#gc
```

```
#gc #full
```

Examples:

```
  #gc
```

```
  #gc #full
```

Returns the same report as Section 7.54 [TOOTSVILLE-USER MEM], page 182,

### 7.33.3 File

Defined in file src/infinity/modern-ops.lisp.

## 7.34 Tootsville-User::Getconfig

### 7.34.1 Function

Getconfig names a function, with lambda list (&REST WORDS):

Reads a configuration key.

All WORDS are expected to be the keywords on the path to the config value.

### 7.34.2 Usage

```
#getconfig PROPERTY
```

```
#getconfig PROP1 PROP2 [...]
```

### 7.34.3 Example

```
#getconfig taskmaster devel
```

Returns the value of the selected configuration property as a string. If the selected property is a key with multiple values (as a property list — plist) associated with it, returns the entire plist (and possibly, nested plists).

### 7.34.4 Changes from 1.2 to 2.0

The format of the configuration file is completely different. The Java properties file has been replaced with a Lisp property list (plist) tree structure which is arranged entirely differently.

### 7.34.5 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.35 Tootsville-User::Getevents

### 7.35.1 Function

Getevents names a function, with lambda list (&REST WORDS):

List GameEvents in your current Zone

UNIMPLEMENTED

Must have staff level 4 (DESIGNER) to use this command.

### 7.35.2 Usage

```
#getevents
```

### 7.35.3 Example

```
#getevents
```

### 7.35.4 See Also

See also Section 7.5 [TOOTSVILLE-USER ADDEVENT], page 132,

### 7.35.5 File

Defined in file src/infinity/legacy-ops.lisp.



## 7.36 Tootsville-User::Getmotd

### 7.36.1 Function

Getmotd names a function, with lambda list (&REST WORDS):

Retrieve the current Message Of The Day as a server message.

### 7.36.2 Usage

```
#getmotd
```

### 7.36.3 Example

```
#getmotd
```

### 7.36.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.37 Tootsville-User::Getschedule

### 7.37.1 Function

Getschedule names a function, with lambda list (&REST WORDS):

Get schedule

Gets all scheduled events in the metronome system, with their schedules.

### 7.37.2 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.38 Tootsville-User::Getschedulefor

### 7.38.1 Function

Getschedulefor names a function, with lambda list (&REST WORDS):

Get scheduled events for a particular class (scheduled by that class)

UNIMPLEMENTED

### 7.38.2 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.39 Tootsville-User::Getuvar

### 7.39.1 Function

Getuvar names a function, with lambda list (&REST WORDS):

Get a user variable.

UNIMPLEMENTED

Must have staff level 4 (DESIGNER) to use this command.

### 7.39.2 Usage

```
#getuvar [LOGIN] [VARIABLE]
```

user name of a character #me for the user you are logged in as

### 7.39.3 Examples

```
#getuvar mouser d
```

```
#getuvar #me d
```

### 7.39.4 See Also

See also Section 7.78 [TOOTSVILLE-USER SETUVAR], page 209, Section 7.40 [TOOTSVILLE-USER GETUVARS], page 167,

### 7.39.5 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.40 Tootsville-User::Getuvars

### 7.40.1 Function

Getuvars names a function, with lambda list (&REST WORDS):

Get all user variables for a given user.

UNIMPLEMENTED

### 7.40.2 Usage

```
#getuvars [LOGIN]
```

```
#getuvars #me
```

### 7.40.3 Examples

```
#getuvars mouser
```

```
#getuvars #me
```

### 7.40.4 See Also

See also Section 7.78 [TOOTSVILLE-USER SETUVAR], page 209, Section 7.39 [TOOTSVILLE-USER GETUVAR], page 166,

### 7.40.5 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.41 Tootsville-User::Getvar

### 7.41.1 Function

Getvar names a function, with lambda list (&REST WORDS):

Get a room variable.

### 7.41.2 Usage

```
#getvar @[ROOM] [VARIABLE]
#getvar [VARIABLE]
```

### 7.41.3 Examples

```
#getvar @tootsSquareWest anim~ropes
#getvar anim~ropes
```

### 7.41.4 See Also

See also Section 7.79 [TOOTSVILLE-USER SETVAR], page 210, Section 7.16 [TOOTSVILLE-USER CLEARVAR], page 143, Section 7.42 [TOOTSVILLE-USER GETVARS], page 169,

### 7.41.5 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.42 Tootsville-User::Getvars

### 7.42.1 Function

Getvars names a function, with lambda list (&REST WORDS):

Get all room variables.

### 7.42.2 Usage

```
#getvars [ROOM]
#getvars
```

### 7.42.3 Examples

```
#getvars tootsSquare
#getvars
```

### 7.42.4 See Also

See also Section 7.79 [TOOTSVILLE-USER SETVAR], page 210, Section 7.16 [TOOTSVILLE-USER CLEARVAR], page 143, Section 7.41 [TOOTSVILLE-USER GETVAR], page 168,

### 7.42.5 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.43 Tootsville-User::Git-Pull

### 7.43.1 Function

Git-Pull names a function, with lambda list (&REST WORDS):

Does a `git pull` in the server directory.

See also Section 7.64 [TOOTSVILLE-USER QUICK-RELOAD], page 195, to actually load any new code that's downloaded.

### 7.43.2 Usage

```
#git-pull
```

Example

```
  #git-pull
```

### 7.43.3 Effects

Sends an admin message with “stand by,” then does a `git pull` in the source directory and returns the results (e.g. “Already up to date.” or otherwise).

### 7.43.4 File

Defined in file `src/infinity/modern-ops.lisp`.



## 7.44 Tootsville-User::Give

### 7.44.1 Function

Give names a function, with lambda list (&REST WORDS):

Give an item as a gift to another user.

### 7.44.2 Usage

```
#give ITEM USER
```

### 7.44.3 Example

```
#give CDCCA838-FB7B-423A-81DA-1514817598DB flappyperry  
UNIMPLEMENTED
```

The item to be gifted must be in your inventory. To give a new item see Section 7.46 [TOOTSVILLE-USER GRANT], page 173,

### 7.44.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.45 Tootsville-User::Givehead

### 7.45.1 Function

Givehead names a function, with lambda list (&REST WORDS):

Grants a new inventory item to a user and equips it.

NOTE: `#grant` and `#givehead` are identical, except that `#givehead` equips the item and `#grant` does not. See also Section 7.46 [TOOTSVILLE-USER GRANT], page 173.

### 7.45.2 Usage

`#givehead` TEMPLATE USER

### 7.45.3 Example

```
#givehead 1337 catvllle
```

This creates a new item from the item template number indicated, and equips it on the recipient. To give a gift from your own inventory, see Section 7.44 [TOOTSVILLE-USER GIVE], page 171. To grant a new item without equipping it, see Section 7.46 [TOOTSVILLE-USER GRANT], page 173.

### 7.45.4 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.46 Tootsville-User::Grant

### 7.46.1 Function

Grant names a function, with lambda list (&REST WORDS):

Grants a new inventory item to a user.

NOTE: `#grant` and `#givehead` are identical, except that `#givehead` equips the item and `#grant` does not. See also Section 7.45 [TOOTSVILLE-USER GIVEHEAD], page 172.

### 7.46.2 Usage

`#grant` TEMPLATE USER

### 7.46.3 Example

```
#grant 1337 catv11e
```

This creates a new item from the item template number indicated, and gives it to the recipient. To give a gift from your own inventory, see Section 7.44 [TOOTSVILLE-USER GIVE], page 171. To grant a new item and equipping it, see Section 7.45 [TOOTSVILLE-USER GIVEHEAD], page 172.

### 7.46.4 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.47 Tootsville-User::Headcount

### 7.47.1 Function

Headcount names a function, with lambda list (&REST WORDS):

Get headcount information about the running system.

### 7.47.2 Usage

```
#headcount #all
#headcount #members
#headcount #room
#headcount #highwater
```

### 7.47.3 Examples

```
#headcount #all
#headcount #members
#headcount #room
#headcount #highwater
```

### 7.47.4 Headcount All

Gives the total number of users online now.

### 7.47.5 Headcount Members

Gives the total number of patron users or builder Toot users online now.

### 7.47.6 Headcount Room

Gives the total number of users who are within “earshot” of the person invoking this command.

### 7.47.7 Headcount Highwater

Gives the high-water mark of the maximum number of simultaneous users who have been online since the last boot.

### 7.47.8 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.48 Tootsville-User::Infinity-Stats

### 7.48.1 Function

Infinity-Stats names a function, with lambda list (&REST WORDS):

Returns some statistics about Infinity-mode requests.

See Section 8.739 [TOOTSVILLE INFINITY-STATS], page 1023.

### 7.48.2 Usage

```
#infinity-stats
```

### 7.48.3 Example

```
#infinity-stats
```

### 7.48.4 File

Defined in file src/infinity/modern-ops.lisp.

## 7.49 Tootsville-User::Inv

### 7.49.1 Function

Inv names a function, with lambda list (&REST WORDS):

Get a user's inventory

Get inventory items for a particular user. By default, this will bring up only the active items — e.g. clothing being worn, and so forth.

### 7.49.2 Usage

To get all active inventory for an user: `#inv LOGIN`

To get all inventory for an user, active or inactive (this may be very long!): `#inv LOGIN #all`

To get inventory of a particular type, active or inactive: `#inv LOGIN #type TYPE`

The type strings accepted are those accepted by Section 8.697 [TOOTSVILLE INFINITY-GET-INVENTORY-BY-TYPE], page 962; this means that both the `$SPECIFIC-TYPE` and `TYPE-SET-NAME` forms are accepted. The list of specific types might include e.g. `$Hair`, and a type-set-name might be something like `clothing`. The set of available type-set-names is specified in the configuration file.

### 7.49.3 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.50 Tootsville-User::Kick

### 7.50.1 Function

Kick names a function, with lambda list (&REST WORDS):

Kick a user offline for a certain reason.

### 7.50.2 Usage

```
#kick [REASONCODE] [LOGIN]
```

Kick LOGIN offline for REASONCODE

```
#kick #list
```

List reason codes.

### 7.50.3 Example

```
#kick bully pil
```

```
#kick #list
```

### 7.50.4 Reason Codes

BULLY      Bullies are not allowed here.  
 CHEAT      Cheaters are not allowed here.  
 DIAMOND    Watch your language around children.  
 MEAN       Don't be mean!  
 NICE        Be nice!  
 PARENT     You need your parent's permission to play in Tootsville.

### 7.50.5 Reason Codes from 1.2

These are no longer supported

PER.MAIL    Don't share personal information like eMail addresses!  
 PER.NAME    Don't share personal information like your real name!  
 PER.PASS    Don't share personal information like passwords!  
 PER.CHAT    Don't share personal information like chat and instant messaging information!  
 PER.LOCA    Don't share personal information like your location!  
 PER.AGES    Don't share personal information like your age!  
 PER.BDAY    Don't share personal information like your birth date!  
 BUL.MEAN    Don't be mean!  
 OBS.RUDE    Don't be rude!  
 OBS.FOUL    Don't use foul words!  
 NET.CHTR    No cheating!

`APP.PARN` You need your parent's permission in order to chat in Tootsville.

`APP.MAIL` You need to confirm your eMail address in order to chat in Tootsville.

`APP.AGES` Lying about your birth date is against the law!

### **7.50.6 File**

Defined in file `src/infinity/legacy-ops.lisp`.



## 7.51 Tootsville-User::King

### 7.51.1 Function

King names a function, with lambda list (&REST WORDS):

Upgrade a user account

### 7.51.2 Usage

```
#king [DAYS] [LOGIN]
```

### 7.51.3 Example

```
#king 2 flappyperry
```

### 7.51.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.52 Tootsville-User::Liftban

### 7.52.1 Function

Liftban names a function, with lambda list (&REST WORDS):

Lift the ban upon a user.

### 7.52.2 Usage

```
#liftban REASON USER yes
```

NOTE: In order to un-ban a user, you must key in the literal word `yes` as the third parameter, and supply the ban reason as the first. This is to avoid accidentally lifting a ban.

### 7.52.3 Example

```
#liftban CHEAT silly-biscuits yes
```

### 7.52.4 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.53 Tootsville-User::Loadlists

### 7.53.1 Function

Loadlists names a function, with lambda list (&REST WORDS):

Reload the censorship lists.

### 7.53.2 Usage

```
#loadlists
```

```
#loadlists #blacklist
```

```
#loadlists #redlist
```

This reloads the blacklist or redlist from the database, discarding any unsaved or recent changes.

### 7.53.3 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.54 Tootsville-User::Mem

### 7.54.1 Function

Mem names a function, with lambda list (&REST WORDS):

Display some memory usage and other debugging type information as an pop-up message.

This is an abbreviated version of the output of ROOM (see the Common Lisp HyperSpec) on the server.

### 7.54.2 Usage

```
#mem
```

### 7.54.3 Example

```
#mem
```

### 7.54.4 Example report

```
This server is Inktomi.  
Dynamic space usage is: 756,315,840 bytes.  
Immobile space usage is: 31,537,408 bytes (134,512 bytes overhead).  
Read-only space usage is: 0 bytes.  
Static space usage is: 1,344 bytes.  
Control stack usage is: 9,656 bytes.  
Binding stack usage is: 832 bytes.  
Control and binding stack usage is for the current thread only.  
Garbage collection is currently enabled.
```

Note that the output of ROOM (see the Common Lisp HyperSpec) can vary wildly depending on the compiler used; the above is from a build of Tootsville compiled under SBCL, which is the expected environment, but there is no guarantee that this will not change in future.

### 7.54.5 Changes from 1.2 to 2.0

In Romance 1, we were running in a Java Virtual Machine (JVM), so the output of mem was quite differently formatted.

### 7.54.6 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.55 Tootsville-User::Metronome

### 7.55.1 Function

Metronome names a function, with lambda list (&REST WORDS):

Display information about or micromanage the metronome.

### 7.55.2 Usage

`#metronome` [OPTION]

### 7.55.3 Examples

```
#metronome #help
#metronome #rate
#metronome #last
#metronome #start
#metronome #stop
#metronome #restart
#metronome #tick
#metronome #list
#metronome #cancel <NAME>
```

#### 7.55.3.1 Options

```
#help      list these options
#rate      Displays a message indicating the rate that the metronome ticks in milliseconds.
           Always 1000 (1s).
#last      Displays a message indicating the time in milliseconds when the last metronome
           tick occurred. Always rounded to 1s.
#start     Starts the metronome.
#stop      Stops the metronome.
#restart   Restarts the metronome.
#tick      Forces the metronome to tick.
#list      List all tasks scheduled with the metronome
#cancel <NAME>
           Cancel a specific task by name
```

#### 7.55.4 Changes from 1.2 to 2.0

Added `#metronome #help`, `#metronome #list`, and `#metronome #cancel NAME`

#### 7.55.5 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.56 Tootsville-User::Motd

### 7.56.1 Function

Motd names a function, with lambda list (&REST WORDS):

Set the message of the day.

### 7.56.2 Usage

```
#motd The new message of the day, literally.
```

### 7.56.3 Example

```
#motd Don't forget that Hallowe'en in Tootsville is on the 30th --- get your costumes
```

The message of the day is echoed to every user as they sign in, before they choose a Toot. It is *not* echoed to children.

### 7.56.4 Changes from 1.2 to 2.0

In Romance II, we do not display the MotD to children, but their parents will see it when approving their sign-on.

### 7.56.5 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.57 Tootsville-User::Mute

### 7.57.1 Function

Mute names a function, with lambda list (&REST WORDS):

Mute a user or area.

This is a simpler form of Section 7.85 [TOOTSVILLE-USER STFU], page 216, that does not accept a duration.

UNIMPLEMENTED

```
#mute user-name
```

The player muted will receive an admin message:

You are no longer allowed to speak in Tootsville.

The invoking user will receive a confirmation.

USER-NAME is no longer allowed to speak in Tootsville.

If the user cannot be found,

Can't find user "USER-NAME"

### 7.57.2 See also

See Also: Section 7.85 [TOOTSVILLE-USER STFU], page 216,

### 7.57.3 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.58 Tootsville-User::Nuke

### 7.58.1 Function

Nuke names a function, with lambda list (&REST WORDS):

Forcibly disconnect everyone in an area.

This is a horrible command and it lies to the user.

Every user who is “near” (see Section 8.927 [TOOTSVILLE NEARP], page 1212) the spot named in this command will be kicked offline with an admin message that lies to them about what has happened.

### 7.58.2 Usage

```
#nuke SPOT-NAME
```

### 7.58.3 Example

```
#nuke Toot-square
```

### 7.58.4 Results

Every user will be given an admin message which is essentially a lie:

A problem with the game caused you to be disconnected. We’re sorry for the inconvenience, and a system operator is already aware of the situation. You can sign back in immediately.

The auto-reconnect code will likely fire off, causing a login storm from all affected users.

### 7.58.5 Rationale

There should be no reason to use this command in Romance 2.0

### 7.58.6 Rationale for version 1.2

In Tootsville IV, there could exist a situation that caused message traffic in a particular room to hang, due to obscure timing bugs that could manifest under stress.

The fastest solution was to simply disconnect everyone in the room, allowing the system to recover.

In real life, this command was used less than once a month, but that was still far too often, and a precise cause for the problem was never narrowed down; the new engine should not have this kind of timing issue.

### 7.58.7 File

Defined in file `src/infinity/legacy-ops.lisp`.



## 7.59 Tootsville-User::Parentapproves

### 7.59.1 Function

Parentapproves names a function, with lambda list (&REST WORDS):

Signal that a parent approves a user signing in.

### 7.59.2 Usage

```
#parentapproves TOOT
```

### 7.59.3 Example

```
#parentapproves Pil
```

### 7.59.4 Limitations

This is only useful if TOOT is a child Toot account has begun to sign in and requested parent permission — that is, there must be a pending child request from TOOT.

### 7.59.5 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.60 Tootsville-User::Ping

### 7.60.1 Function

Ping names a function, with lambda list (&REST WORDS):

Ping the server, to force a neutral administrative message reply.

### 7.60.2 Usage

```
#ping
```

### 7.60.3 Example

```
#ping
```

### 7.60.4 Reply

The reply is an administrative message saying simply:

```
Pong!
```

### 7.60.5 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.61 Tootsville-User::Place

### 7.61.1 Function

Place names a function, with lambda list (&REST WORDS):

Put a thing or a Place into the game

“Place a thing” or “create a place” in the game.

This command supports the basic types of event Places, and adds them to the room in the given WHERE place. WHERE can be a diamond-shaped area around the operator issuing the command (using #here, #here-tiny, or #here-big), or can be an explicitly-issued polygon string or circle designator. The event region ID will be automatically assigned.

These are usually communicated to the client as “room variables;” see Section 8.702 [TOOTSVILLE INFINITY-GET-ROOM-VARS], page 969, for a description of that protocol.

UNIMPLEMENTED. Target version: 0.7

### 7.61.2 Usage

```
#place #list
#place WHERE #download ITEM-TEMPLATE-NUMBER URL [FACING]
#place WHERE #exit MONIKER
#place WHERE #fountain ITEM-TEMPLATE-NUMBER
#place WHERE #game GAME-MONIKER GAME-ATTRIBUTES
#place WHERE #item ITEM-TEMPLATE-NUMBER [FACING]
#place WHERE #item2 ITEM-TEMPLATE-NUMBER OTHER-ITEM-TEMPLATE-NUMBER
#place WHERE #mini MINIGAME-MONIKER
#place WHERE #place PLACE-KIND
#place WHERE #room MONIKER
#place WHERE #shop ITEM-TEMPLATE-NUMBER PRICE [FACING]
#place WHERE #snowball ITEM-TEMPLATE-NUMBER [FACING]
#place WHERE #unwalk
#place WHERE #vitem ITEM-TEMPLATE-NUMBER [FACING]
#place WHERE #walk
```

#place #list will give a brief reminder of the #place subcommand syntax, although this manual should be considered the canonical reference source.

WHERE can be one of:

**#here**      The location of the operator issuing the command, surrounded by an “average size” polygon approximating a circle.

**#here-tiny**      The location of the operator issuing the command, surrounded by a “tiny” polygon approximating a circle.

**#here-big**      The location of the operator issuing the command, surrounded by a “big” polygon approximating a circle.

**A point**      An x,z coordinate pair; e.g. 100,100.

**A polygon designator**

The specific location can be outlined as a series of x,z coördinates delimited by tildes; e.g. 100,100~100,200~200,200~200,100.

**A circle designator A polygon approximating a circle originating**

at the location of the operator issuing the command, whose radius R and number of segments S are specified, delimited by a x character; e.g. 100x10. A circle can be designated to start at another position with the notation 50,60+100x10, where the center will be at x position 50, z position 60.

The second parameter indicates the sort of thing that will be added, as enumerated in the following sections.

**7.61.3 Examples**

```
#place #here #item 1337
#place #here #room TootSweets
#place #here #vitem 42
#place #here-tiny #exit TootSquare
#place #here #mini Minigame.js minigame
#place #here-big #walk
#place 100,100~100,200~200,200~200,100 #unwalk
#place 100x10 #item 1234
```

**7.61.4 #download Placing a download trigger item**

An item will be placed at the point specified, which must be a single point (or a named spot).

**7.61.5 #exit Placing a transwarp conduit**

An “exit” is a hyperspace link between two spots in the game universe. A place will be created at the indicated location. Any player who enters into the “exit” place will be immediately relocated to the named spot indicated.

**7.61.5.1 Changes from 1.2 to 2.0**

In Romance 1.2, exits were linked between rooms. A specially-formatted exit designator could indicate to which exit (entrance) in the other room to link the player.

In Romance 2.0, the exit’s destination is an arbitrary point with a “spot” name assigned to it.

**7.61.6 #fountain Placing a magic fountain**

WRITEME

**7.61.7 item Placing an item**

A furniture item will be placed at the position indicated. Position must be #here or a point coördinate pair. The identifier is an item template ID number. An instance of the item will be placed at that point. An optional facing direction can be specified, either in radians, or from the set N NE E SE S SW W NW.

### 7.61.7.1 Changes from 1.2 to 2.0

In Romance 1.2, this was used to place an item-gifting spot, which was invisible.

### 7.61.8 #item2

This is no longer supported in Romance 2.0.

### 7.61.8.1 Changes from 1.2 to 2.0

In Romance 1.2, `item2` provided different items to paid (“V.I.T.”) or unpaid (regular) players. Since Tootsville V is free to play, this is no longer needed.

### 7.61.9 #place Placing a Place designator

A Place designator `WRITEME`

### 7.61.10 room Placing a “room” (spot) marker

A “spot” designator will be created at the point indicated, which must be `#here` or a point coordinate pair. The moniker given will be associated with the spot and can be used for certain other commands.

### 7.61.11 #shop Placing a shop item

A shop item is placed at the position indicated. Position must be `#here` or a point coordinate pair. The identifier is an item template ID number. An instance of the item will be placed at that point. An optional facing direction can be specified, either in radians, or from the set `N NE E SE S SW W NW`. Any player who clicks on the item at this spot will receive a prompt offering to sell them the item at the price indicated.

The price given is in peanuts, unless it begins with the letter `F`, in which case it is given in fairy dust.

### 7.61.12 #mini Placing a minigame

Minigames are not supported in Romance 2.0, although they could return in some form. In-world games based upon Section 8.692 [TOOTSVILLE INFINITY-GAME-ACTION], page 954, are supported still, q.v. See Also `#place #game` in this section.

### 7.61.12.1 Changes from 1.2 to 2.0

In Tootsville IV, minigames were Adobe Flash applets which were loaded into the main game environment. This is no longer the case.

### 7.61.13 #snowball Placing a snowball source pile

`WRITEME`

### 7.61.14 #unwalk Placing an unwalkable space

An unwalkable space is a specific type of Place designator given special consideration. It presents as a very tall invisible object that blocks navigation.

Unwalkable spaces are *not* currently supported by the Tootsville V client software.

To remove an unwalkable space, place a `#walk` space that covers it.

### 7.61.15 `vitem` Placing an item-gifting item

An item-gifting spot will be placed at the position indicated. Position must be `#here` or a point coördinate pair. The identifier is an item template ID number. An instance of the item will be placed at that point. An optional facing direction can be specified, either in radians, or from the set N NE E SE S SW W NW. Any player who clicks on the item at this spot will receive an instance of the template in their inventory, and a friendly pop-up message with a description of the item. Only one item per player will be given.

#### 7.61.15.1 Changes from 1.2 to 2.0

In Romance 1.2, `vitem` gifts were only for “V.I.T.” (paid) players, and `item` was for everyone. `item` has been repurposed for furniture placement. Also, item gifting spots were invisible and triggered by the player walking into them.

### 7.61.16 `#walk` Placing a walkable space

This designates that the place specified is walkable space; if it intersects any place previously marked as unwalkable, it will be reverted.

#### 7.61.17 Implementation note

Each subcommand is implemented by a “private” function named `%OPERATOR-PLACE-subcommand` in the Tootsville package.

### 7.61.18 File

Defined in file `src/infinity/legacy-ops.lisp`.

## **7.62 Tootsville-User::Purgephysics**

### **7.62.1 Function**

Purgephysics names a function, with lambda list (&REST WORDS):

Purge pending physics interactions.

This is a no-op.

### **7.62.2 Changes from 1.2 to 2.0**

In Romance II, physics are handled by the clients. This command is no longer needed.

### **7.62.3 File**

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.63 Tootsville-User::Push-Script

### 7.63.1 Function

Push-Script names a function, with lambda list (&REST WORDS):

Instruct clients to load a new script file.

Pushes a script filename to clients. The pathname must be relative to the `play.tootsville.org` server (or its equivalent in other clusters). Used to push an emergency software update without requiring players to reload.

Pushes a script filename to clients. The pathname must be relative to the `play.tootsville.org` server (or its equivalent in other clusters). Used to push an emergency software update without requiring players to reload.

### 7.63.2 Usage

```
#push-script PATHNAME
```

### 7.63.3 Example

```
#push-script /play/UI/UI.js
```

### 7.63.4 File

Defined in file `src/infinity/modern-ops.lisp`.



## 7.64 Tootsville-User::Quick-Reload

### 7.64.1 Function

Quick-Reload names a function, with lambda list (&REST WORDS):

Quicklisp reload of the Tootsville package from disk.

Reloads the ASDF file with ASDF::LOAD-ASD (not in this manual) and then does a QL::QUICKLOAD (not in this manual). See Section 8.1059 [TOOTSVILLE RELOAD-PRODUCTION], page 1344.

### 7.64.2 Usage

`#quick-reload`

Example:

```
#quick-reload
```

### 7.64.3 Effects

Sends an admin message with “Stand by,” then calls Section 8.1059 [TOOTSVILLE RELOAD-PRODUCTION], page 1344, (qv). When completed, reports the version of the ASDF component now loaded (i.e. the version number from `tootsville.asd`).

### 7.64.4 File

Defined in file `src/infinity/modern-ops.lisp`.

## 7.65 Tootsville-User::Rc

### 7.65.1 Function

Rc names a function, with lambda list (&REST WORDS):

Run an RC (Run Commands) script.

UNIMPLEMENTED

Run an RC (RunCommands) script. Both the “system run commands” (“run”) method and the “new zone run commands” (“newZone”) method will be executed; the

### 7.65.2 Usage

```
#rc
```

### 7.65.3 Example

```
#rc
```

```
WRITEME
```

### 7.65.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.66 Tootsville-User::Reboot

### 7.66.1 Function

Reboot names a function, with lambda list (&REST WORDS):

Restart the game server.

No, really; this actually kills the game server with an error exit so that it will (hopefully) be restarted by SystemD.

This is a violent way to go, and is for emergencies *only*.

### 7.66.2 Usage

```
#reboot
```

### 7.66.3 Example

```
#reboot
```

### 7.66.4 Actual Effects

The server will quit with Unix exit status 66 in 3 seconds after the command is received.

### 7.66.5 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.67 Tootsville-User::Reloadconfig

### 7.67.1 Function

Reloadconfig names a function, with lambda list (&REST WORDS):

Reloads configuration properties.

### 7.67.2 Usage

```
#reloadconfig
```

### 7.67.3 Example

```
#reloadconfig
```

### 7.67.4 Effect

Reloads the configuration file (`.config/Tootsville/Tootsville.config.lisp` under the server owner's home directory). See Section 8.817 [TOOTSVILLE LOAD-CONFIG], page 1102. Reports back the file loaded, and the author and write date of the file.

### 7.67.5 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.68 Tootsville-User::Retire

### 7.68.1 Function

Retire names a function, with lambda list (&REST WORDS):

Retire a server.

Forces a server to retire. This will disconnect anyone currently connected via WebSockets to that server; they should reconnect through the load balancer. Use `#evacuate` to first move users to another server (see Section 7.28 [TOOTSVILLE-USER EVACUATE], page 155).

### 7.68.2 Usage

```
#retire SERVER  
#retire
```

### 7.68.3 Examples

```
#retire game3.test.Tootsville.org  
#retire
```

### 7.68.4 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.69 Tootsville-User::Run

### 7.69.1 Function

Run names a function, with lambda list (&REST WORDS):

Run an arbitrary nullary Lisp function or method

### 7.69.2 USave

```
#run FUNCTION
```

```
#run PACKAGE FUNCTION
```

### 7.69.3 Examples

```
#run ws-stats
```

```
#run infinity-stats
```

```
#run sb-ext quit
```

### 7.69.4 Changes from 1.2 to 2.0

In 1.x: Run an arbitrary Java routine through an uploaded Runnable or RunCommands class

In 2.x: Run arbitrary nullary Lisp function or method

### 7.69.5 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.70 Tootsville-User::Saveroomvars

### 7.70.1 Function

Saveroomvars names a function, with lambda list (&REST WORDS):

Save room variables.

No longer used.

### 7.70.2 Legacy Usage (1.2)

In Romance 1.2, this would save the effective room variables in a room to the database permanently. Now, all things that room variables used to represent are already persisted to the database.

### 7.70.3 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.71 Tootsville-User::Scotty

### 7.71.1 Function

Scotty names a function, with lambda list (&REST WORDS):

Force a user to relocate to another location

### 7.71.2 Usage

```
#scotty TOOT NAMED-PLACE
```

```
#scotty TOOT LATITUDE LONGITUDE [ALTITUDE] [WORLD]
```

Altitude is optional and defaults to 0.

World is optional and defaults to CHOR.

### 7.71.3 Examples

```
#scotty mouser TootSquareWest
```

```
#scotty mouser -1 0 0 CHOR
```

```
#scotty mouser -1 0
```

### 7.71.4 Changes from 1.2 to 2.0

In 1.2, this moved an user into another room.

Usage: #scotty [LOGIN] [ROOM]

Examples: #scotty mouser tootSquareWest

### 7.71.5 File

Defined in file src/infinity/legacy-ops.lisp.



## 7.72 Tootsville-User::Script

### 7.72.1 Function

Script names a function, with lambda list (&REST WORDS):

Push a new function into the TOOTSVILLE-USER package.

UNIMPLEMENTED.

### 7.72.2 Usage

```
#script TITLE SOURCE TEXT ...
```

### 7.72.3 Example

```
#script simply-string "simply"
```

### 7.72.4 File

Defined in file src/infinity/modern-ops.lisp.

## 7.73 Tootsville-User::Server-List

### 7.73.1 Function

Server-List names a function, with lambda list (&REST WORDS):

Enumerate the servers active in this cluster.

See Section 8.1120 [TOOTSVILLE SERVER-LIST], page 1405,

### 7.73.2 Usage

```
#server-list
```

### 7.73.3 Example

```
#server-list
```

### 7.73.4 File

Defined in file src/infinity/modern-ops.lisp.

## 7.74 Tootsville-User::Setavatarcolors

### 7.74.1 Function

Setavatarcolors names a function, with lambda list (&REST WORDS):

Sets the base and extra colors of a user's avatar.

### 7.74.2 Usage

```
#setavatarcolors LOGIN BASE EXTRA
```

Each of BASE and EXTRA can be specified in a number of formats.

- CSS Style uses a decimal triplet in the form `rgb(r,g,b)` — identified by the literal string `rgb`. Each of R, G, and B are in the range 0 to 255.
- HTML Style uses a # sign plus either 3 or 6 hex characters, in the form `#rgb` or `#rrggbb`. The # sign is optional.
- Named colors are supported as per Section 8.945 [TOOTSVILLE PARSE-COLOR24], page 1230,

### 7.74.3 Examples

```
#setavatarcolors mouser #000000 #ffffff  
#setavatarcolors mouser rgb(0,0,0) rgb(255,255,255)
```

See also Section 7.21 [TOOTSVILLE-USER DOODLE], page 148, for a similar-but-different way to set avatar colors.

### 7.74.4 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.75 Tootsville-User::Setbadge

### 7.75.1 Function

Setbadge names a function, with lambda list (&REST WORDS):

Set the badge on a map area.

### 7.75.2 Usage

```
#setbadge  
#setbadge BADGE MONIKER  
#setbadge #me MONIKER  
#setbadge BADGE #here  
#setbadge #me #here
```

NOTE: Using #setbadge with no parameters will assume default values which are identical to typing #setbadge #me #here

### 7.75.3 Examples

```
#setbadge snowcone TootSquareWest  
#setbadge #me TootSquare  
#setbadge snowcone #here  
#setbadge #me #here
```

### 7.75.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.76 Tootsville-User::Setconfig

### 7.76.1 Function

Setconfig names a function, with lambda list (&REST WORDS):

Set a config property.

### 7.76.2 Usage

```
#setconfig PROPERTY VALUE
```

```
#setconfig PROP1 PROP2 VALUE
```

PROPERTY is a sequence of keywords, which must be delimited by spaces. Omit the leading `:` on the keyword names.

### 7.76.3 Example

```
#setconfig rollbar access-token 1234567890
```

Changes made with this command are only effective until the configuration file is reloaded. See Section 7.67 [TOOTSVILLE-USER RELOADCONFIG], page 198, and Section 8.817 [TOOTSVILLE LOAD-CONFIG], page 1102.

### 7.76.4 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.77 Tootsville-User::Setstafflevel

### 7.77.1 Function

Setstafflevel names a function, with lambda list (&REST WORDS):

Set the staff level for a user

UNIMPLEMENTED

WRITEME

### 7.77.2 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.78 Tootsville-User::Setuvar

### 7.78.1 Function

Setuvar names a function, with lambda list (&REST WORDS):

Set a user variable.

UNIMPLEMENTED

Set a user variable. Must have staff level 4 (DESIGNER) to use this command.

### 7.78.2 Usage

```
#setuvar [@[LOGIN] VARIABLE [=] VALUE...
#setuvar @LOGIN VARIABLE=VALUE
#setuvar @LOGIN VARIABLE = VALUE
#setuvar @LOGIN VARIABLE VALUE
#setuvar VARIABLE=VALUE
#setuvar VARIABLE = VALUE
#setuvar VARIABLE VALUE
```

NOTE: Using #setuvar without an @[LOGIN] parameter will apply the changes to the user issuing the command.

### 7.78.3 Example

```
#setuvar @mouser d = 254~376~254~376~SE~1267735566759
#setuvar d 254~376~254~376~SE~1267735566759
```

See Section 8.733 [TOOTSVILLE INFINITY-SET-USER-VAR], page 1012, for a discussion of supported user variables in Romance II.

### 7.78.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.79 Tootsville-User::Setvar

### 7.79.1 Function

Setvar names a function, with lambda list (&REST WORDS):

Set a room variable.

UNIMPLEMENTED.

This used to be used to set Room Variables, which were the main way that the game design worked in Tootsville IV. This was largely automated through Eric Feiling’s “Zookeeper” application.

In Tootsville V, however, room variables are a reflection of the underlying database structures and are automatically generated as needed; there is not currently a way to backwards-supply the variables’ data.

### 7.79.2 Description from Romance 1.2

Set a room variable. Must have staff level 4 (DESIGNER) to use this command.

Usage

```
#setvar #replace [@@ROOM] VARIABLE FIND REPLACE
```

```
#setvar [@@ROOM] VARIABLE VALUE...
```

**WARNING: SETTING ROOM VARIABLES TO INVALID VALUES CAN CAUSE UNEXPECTED RESULTS. DOUBLE CHECK ALL VALUES BEING SET FOR CORRECTNESS.**

Use #replace to change a room variable from one value to another.

### 7.79.3 Examples

```
#setvar @@tootsSquareWest anim~ropes 2
```

```
#setvar anim~ropes 2
```

### 7.79.4 File

Defined in file src/infinity/legacy-ops.lisp.



## 7.80 Tootsville-User::Shanghai

### 7.80.1 Function

Shanghai names a function, with lambda list (&REST WORDS):

Force a client into a different room and zone

UNIMPLEMENTED

WRITEME

### 7.80.2 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.81 Tootsville-User::Shout

### 7.81.1 Function

Shout names a function, with lambda list (&REST WORDS):

Speak in another zone.

This is intended for using operator commands in a remote zone, not normal chat messages.

Since there are no longer zones, this command is not currently supported. The command name may be re-used for sending commands to a different server in future.

### 7.81.2 Usage

```
#shout [ZONE] [ROOM] [COMMAND...]
```

### 7.81.3 Examples

```
#shout dottie tootSquareWest #wall Hello Everyone
```

```
#shout dottie tootSquare #retire
```

See modern version Section 7.8 [TOOTSVILLE-USER AT], page 135, also

### 7.81.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.82 Tootsville-User::Spawnroom

### 7.82.1 Function

Spawnroom names a function, with lambda list (&REST WORDS):

Mark a “spot” in the game.

UNIMPLEMENTED.

Mark the current position of your Toot as a named “spot” in the game world.

### 7.82.2 Usage

```
#spawnroom [MONIKER]
```

### 7.82.3 Changes from 1.2 to 2.0

This command has been completely repurposed, since there are no longer rooms.

### 7.82.4 Legacy 1.2 Documentation

Create a new room in the current zone.

### 7.82.5 Usage in 1.2

```
#spawnroom [MONIKER] [TITLE] [SWF]
```

```
#spawnroom [MONIKER] [TITLE]
```

NOTE: Uses tootCastleJoust.swf as default. This can be set after the room has been created by setting the 'f' room variable.

### 7.82.6 Examples of 1.2 syntax

```
#spawnroom tootCastleJoust2 Joust2 tootCastleJoust.swf
```

```
#spawnroom tootCastleJoust2 Joust2
```

### 7.82.7 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.83 Tootsville-User::Spawnzone

### 7.83.1 Function

Spawnzone names a function, with lambda list (&REST WORDS):

Create a new zone.

### 7.83.2 Usage

```
#spawnzone [ZONE]
```

### 7.83.3 Examples

```
#spawnzone Cupcake  
WRITEME
```

### 7.83.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.84 Tootsville-User::Speak

### 7.84.1 Function

Speak names a function, with lambda list (&REST WORDS):

Allows a user to speak who had previously been muted.

See Section 7.57 [TOOTSVILLE-USER MUTE], page 185, Section 7.85 [TOOTSVILLE-USER STFU], page 216, for ways to mute a character.

### 7.84.2 Usage

```
#speak [LOGIN]
```

### 7.84.3 Examples

```
#speak flappyperry
```

### 7.84.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.85 Tootsville-User::Stfu

### 7.85.1 Function

Stfu names a function, with lambda list (&REST WORDS):

Silences (mutes) a user.

### 7.85.2 Usage

```
#stfu TOOT
```

```
#stfu TOOT MINUTES
```

### 7.85.3 Example

```
#stfu louis
```

```
#stfu louis 30
```

### 7.85.4 Effects

This sets an attribute on TOOT that prevents them from actually sending any public speech messages; however, *that user will not know*. The user will see their own speech, but it will not be echoed to anyone else.

In other words, this basically sets up a global ignore of the user to whom it is applied.

If no time limit is given, it is effective for 24 Earth hours (1,440 Earth minutes).

See also: Section 7.57 [TOOTSVILLE-USER MUTE], page 185, for a more direct form that does not have a fixed duration.

### 7.85.5 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.86 Tootsville-User::Testcensor

### 7.86.1 Function

Testcensor names a function, with lambda list (&REST WORDS):

Test a message with the censor, displays the filter result.

UNIMPLEMENTED.

### 7.86.2 Usage

```
#testcensor [MESSAGE]
```

### 7.86.3 Examples

```
#testcensor This message will be filtered and the result will be displayed.
```

### 7.86.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.87 Tootsville-User::Unbuild

### 7.87.1 Function

Unbuild names a function, with lambda list (&REST WORDS):

Destroy a named spot.

UNIMPLEMENTED.

Destroys a named spot.

### 7.87.2 Usage

```
#unbuild ROOM
```

### 7.87.3 Example

```
#unbuild tootUniversity
```

### 7.87.4 Changes from 1.2 to 2.0

In Romance 1.2, this command was used to destroy a room. We no longer have rooms, so it is instead used to destroy named spots.

### 7.87.5 File

Defined in file `src/infinity/legacy-ops.lisp`.



## 7.88 Tootsville-User::Uptime

### 7.88.1 Function

Uptime names a function, with lambda list (&REST WORDS):

    Gives the uptime of the server software.

### 7.88.2 File

Defined in file src/infinity/modern-ops.lisp.

## 7.89 Tootsville-User::V

### 7.89.1 Function

V names a function, with lambda list (&REST WORDS):

Forces a user to say a message.

Mnemonic: Ventriloquism

### 7.89.2 Usage

```
#v LOGIN MESSAGE...
```

### 7.89.3 Example

```
#v mayor-louis I like to cause trouble in Tootsville
```

### 7.89.4 See also

See Section 8.736 [TOOTSVILLE INFINITY-SPEAK], page 1016,

### 7.89.5 Changes from 1.2 to 2.0

This no longer allows ventriloquism of operator commands &c.

### 7.89.6 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.90 Tootsville-User::Verbosebugs

### 7.90.1 Function

Verbosebugs names a function, with lambda list (&REST WORDS):

Set verbose bug backtrace reporting on or off.

UNIMPLEMENTED.

### 7.90.2 Usage

```
#verbosebugs true
```

```
#verbosebugs false
```

### 7.90.3 Impact

When verbose bug reporting is enabled, the user requesting it will receive stack backtraces from unhandled errors as admin messages.

### 7.90.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.91 Tootsville-User::Wall

### 7.91.1 Function

Wall names a function, with lambda list (&REST WORDS):

Write to all players.

Sends an admin (parrot) pop-up message to everyone currently online.

### 7.91.2 Usage

`#wall MESSAGE...`

### 7.91.3 Example

`#wall This message will go to everyone currently on-line.`

### 7.91.4 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.92 Tootsville-User::Wallops

### 7.92.1 Function

Wallops names a function, with lambda list (&REST WORDS):

Write to all operators

Sends an pop-up message to all Builder Toots currently online

### 7.92.2 Usage

#wallops MESSAGE

### 7.92.3 Exampleyy

#wallops This message will go to all other staff members in this zone.

### 7.92.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.93 Tootsville-User::Wallzones

### 7.93.1 Function

Wallzones names a function, with lambda list (&REST WORDS):

Write to all zones.

This is now the same as Section 7.91 [TOOTSVILLE-USER WALL], page 222, qv.

### 7.93.2 Usage

```
#wallzones [MESSAGE...]
```

### 7.93.3 Example

```
#wallzones This message will go to everyone in every zone.
```

### 7.93.4 Changes from 1.2 to 2.0

In Romance 1.2, Zones (shards) were implemented, although not actually used by Tootsville IV. This command wrote to all users in all zones.

### 7.93.5 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.94 Tootsville-User::Whatis

### 7.94.1 Function

Whatis names a function, with lambda list (&REST WORDS):

Displays information about an item template.

The item template info is essentially that which is available from Section 8.787 [TOOTSVILLE ITEM-TEMPLATE-INFO], page 1072.

### 7.94.2 Usage

```
#whatis ITEM-TEMPLATE-ID
```

### 7.94.3 Example

```
#whatis 1337
```

### 7.94.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.95 Tootsville-User::Whereami

### 7.95.1 Function

Whereami names a function, with lambda list (&REST WORDS):

Return an administrative message with the name of the server to which the player is currently connected.

the player is currently connected.

### 7.95.2 Usage

#whereami

### 7.95.3 Example

#whereami

The response admin message is simply the machine name to which you are connected.

### 7.95.4 File

Defined in file src/infinity/legacy-ops.lisp.



## 7.96 Tootsville-User::Whereis

### 7.96.1 Function

Whereis names a function, with lambda list (&REST WORDS):

Locate a user in the game world.

Find out in what room a character is standing, if s/he is logged in at the moment.

Must have staff level 2 (MODERATOR) to use this command.

WRITEME these instructions have not been adapted to Romance II yet.

### 7.96.2 Usage

```
#whereis [LOGIN]
#whereis #everyone
#whereis @[ROOM]
```

User Name of a specific user; #everyone for a the location of every user in the zone;  
@[ROOM] for the location of every user in the specified room.

### 7.96.3 Examples

```
#whereis snowcone
#whereis #everyone
#whereis @tootSquare
```

### 7.96.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.97 Tootsville-User::Who

### 7.97.1 Function

Who names a function, with lambda list (&REST WORDS):

Displays a list of everyone currently in a room.

WRITEME this has not been updated for Romance II yet.

### 7.97.2 Usage

```
#who [ROOM]
```

```
#who
```

NOTE: Leaving off the ROOM parameter will default to displaying for the room the command was initialized in.

### 7.97.3 Examples

```
#who tootSquare
```

```
#who
```

### 7.97.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.98 Tootsville-User::Whoami

### 7.98.1 Function

Whoami names a function, with lambda list (&REST WORDS):

Cause the character to speak his/her name in the current room.

Appears as dialogue in the form: “Hello, my name is NAME.”

Note that the response is public speech; everyone in the room will see it.

### 7.98.2 Usage

```
#whoami
```

Note that the response is public speech; everyone in the room will see it.

### 7.98.3 Example

```
#whoami
```

```
    Hello, my name is Pil.
```

### 7.98.4 File

Defined in file src/infinity/legacy-ops.lisp.

## 7.99 Tootsville-User::Whoareyou

### 7.99.1 Function

Whoareyou names a function, with lambda list (&REST WORDS):

Ask the server who it is.

This command should return version information on some of the critical components used in the game server.

### 7.99.2 Usage

```
#whoareyou
```

### 7.99.3 Example

```
#whoareyou
```

### 7.99.4 Example Response

```
This server is Inktomi, a X86-64 Intel(R) Core(TM) i7 CPU 860 2.80GHz  
running Linux 5.6.8-300.fc32.x86_64 with SBCL 2.0.1-1.fc32. Quicklisp dist  
version 2020-04-27; Ultralisp dist version 20200501011006; Tootsville version  
0.6.4
```

### 7.99.5 Changes from 1.2 to 2.0

The format of the response is different, but the purpose of the command is the same.

### 7.99.6 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 7.100 Tootsville-User::Ws-Bandwidth-By-Source

### 7.100.1 Function

Ws-Bandwidth-By-Source names a function, with lambda list (&REST WORDS):

Returns some statistics about WebSockets bandwidth by source.

See Section 8.1389 [TOOTSVILLE WS-BANDWIDTH-BY-SOURCE], page 1676.

### 7.100.2 Usage

`#ws-bandwidth-by-source`

### 7.100.3 Example

`#ws-bandwidth-by-source`

### 7.100.4 File

Defined in file `src/infinity/modern-ops.lisp`.

## 7.101 Tootsville-User::Ws-Stats

### 7.101.1 Function

Ws-Stats names a function, with lambda list (&REST WORDS):

Returns some statistics about WebSockets connections.

See Section 8.1400 [TOOTSVILLE WS-STATS], page 1687.

### 7.101.2 Usage

```
#ws-stats
```

### 7.101.3 Example

```
#ws-stats
```

### 7.101.4 File

Defined in file src/infinity/modern-ops.lisp.

## 7.102 Tootsville-User::Zoom

### 7.102.1 Function

Zoom names a function, with lambda list (&REST WORDS):

Set the visual Zoom level of a room.

UNIMPLEMENTED

### 7.102.2 Changes from 1.2 to 2.0

In Tootsville IV, rooms could have a different “zoom level” setting the relative size of the display. This is no longer supported; in Tootsville V, the world is a continuous 3D environment.

### 7.102.3 File

Defined in file src/infinity/legacy-ops.lisp.





## 8 Package Tootsville

Let's make some noise!

The Tootsville package is the main container for all Tootsville V (Romance II) server functions.

In future, some of the concerns (e.g. specific facilities like the friendly database accessors or the metronome) may be broken out into their own packages, but for simplicity of early development, everything is in this one big ball of mud.

## 8.1 Tootsville::%Operator-Place-Download

### 8.1.1 Function

%Operator-Place-Download names an undocumented function, with lambda list (WHERE PARAMS).

### 8.1.2 File

Defined in file src/infinity/legacy-ops.lisp.

## 8.2 Tootsville::`%Operator-Place-Exit`

### 8.2.1 Function

`%Operator-Place-Exit` names an undocumented function, with lambda list (WHERE PARAMS).

### 8.2.2 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 8.3 Tootsville::`%Operator-Place-Fountain`

### 8.3.1 Function

`%Operator-Place-Fountain` names an undocumented function, with lambda list (WHERE PARAMS).

### 8.3.2 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 8.4 Tootsville::`%Operator-Place-Game`

### 8.4.1 Function

`%Operator-Place-Game` names an undocumented function, with lambda list (WHERE PARAMS).

### 8.4.2 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 8.5 Tootsville::`%Operator-Place-Item`

### 8.5.1 Function

`%Operator-Place-Item` names an undocumented function, with lambda list (WHERE PARAMS).

### 8.5.2 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 8.6 Tootsville::`%Operator-Place-Mini`

### 8.6.1 Function

`%Operator-Place-Mini` names an undocumented function, with lambda list (WHERE PARAMS).

### 8.6.2 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 8.7 Tootsville::`%Operator-Place-Place`

### 8.7.1 Function

`%Operator-Place-Place` names an undocumented function, with lambda list (WHERE PARAMS).

### 8.7.2 File

Defined in file `src/infinity/legacy-ops.lisp`.



## 8.8 Tootsville::`%Operator-Place-Room`

### 8.8.1 Function

`%Operator-Place-Room` names an undocumented function, with lambda list (WHERE PARAMS).

### 8.8.2 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 8.9 Tootsville::%Operator-Place-Shop

### 8.9.1 Function

%Operator-Place-Shop names an undocumented function, with lambda list (WHERE PARAMS).

### 8.9.2 File

Defined in file src/infinity/legacy-ops.lisp.

## 8.10 Tootsville::`%Operator-Place-Snowball`

### 8.10.1 Function

`%Operator-Place-Snowball` names an undocumented function, with lambda list (WHERE PARAMS).

### 8.10.2 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 8.11 Tootsville::`%Operator-Place-Unwalk`

### 8.11.1 Function

`%Operator-Place-Unwalk` names an undocumented function, with lambda list (WHERE PARAMS).

### 8.11.2 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 8.12 Tootsville::`%Operator-Place-Vitem`

### 8.12.1 Function

`%Operator-Place-Vitem` names an undocumented function, with lambda list (WHERE PARAMS).

### 8.12.2 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 8.13 Tootsville::`%Operator-Place-Walk`

### 8.13.1 Function

`%Operator-Place-Walk` names an undocumented function, with lambda list (WHERE PARAMS).

### 8.13.2 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 8.14 Tootsville::`%Parse-Operator-Place-Where`

### 8.14.1 Function

`%Parse-Operator-Place-Where` names an undocumented function, with lambda list (WHERE).

### 8.14.2 File

Defined in file `src/infinity/legacy-ops.lisp`.

## 8.15 Tootsville::**\*403.Json-Bytes\***

### 8.15.1 Variable

**\*403.Json-Bytes\*** names an undocumented variable with the value of type (SIMPLE-ARRAY (UNSIGNED-BYTE 8) (126))



## **8.16 Tootsville::**\*Acceptors\*****

### **8.16.1 Variable**

**\*Acceptors\*** names a variable:

The set of listening acceptors awaiting incoming connections.

Its value is NIL

## 8.17 Tootsville::**\*Application-Root\***

### 8.17.1 Variable

**\*Application-Root\*** names a variable:

The location in which the application source code is installed.

Its value is of type `PATHNAME`

## 8.18 Tootsville::**\*Async-Channel\***

### 8.18.1 Variable

**\*Async-Channel\*** names a variable:

An LPARALLEL channel used for running asynchronous tasks.

Its value is NIL

## 8.19 Tootsville::**\*Async-Tasks\***

### 8.19.1 Variable

**\*Async-Tasks\*** names a variable:

An LPARALLEL kernel used for running asynchronous tasks.

Its value is NIL

## 8.20 Tootsville::**\*Banhammer\***

### 8.20.1 Variable

**\*Banhammer\*** names a variable:

A list of IP addresses which are banned from connecting.

Its value is the hash table:

## 8.21 Tootsville::**\*Build-Date\***

### 8.21.1 Variable

**\*Build-Date\*** names a variable:

A string representing the year, month, and day at which the program was compiled.

Its value is "2021-2-1"

## 8.22 Tootsville::**\*Cassandra-Blacklist\***

### 8.22.1 Variable

**\*Cassandra-Blacklist\*** names a variable:

The blacklist for text filtering.

This list is applied whenever children or sensitive players are around.

The keys are the string versions of the regexes; the values are the compiled scanners.

Its value is the hash table:

## 8.23 Tootsville::**\*Cassandra-Redlist\***

### 8.23.1 Variable

**\*Cassandra-Redlist\*** names a variable:

The redlist for text filtering.

This list is applied in all areas except "adults only" zones.

The keys are the string versions of the regexes; the values are the compiled scanners.

Its value is the hash table:



## 8.24 Tootsville::**\*Client\***

### 8.24.1 Variable

**\*Client\*** names a variable:

The currently-active client session.

Its value is NIL

## 8.25 Tootsville::**\*Cluster\***

### 8.25.1 Variable

**\*Cluster\*** names a variable:

Cache for Section 8.214 [TOOTSVILLE CLUSTER], page 468, (qv)

Its value is NIL

## 8.26 Tootsville::**\*Compilation\***

### 8.26.1 Variable

**\*Compilation\*** names a variable:

This is used as a temporary output buffer during some maintenance tasks.

Its value is of type SB-IMPL::CHARACTER-STRING-OSTREAM

## 8.27 Tootsville::**\*Compiled\***

### 8.27.1 Variable

**\*Compiled\*** names a variable:

A string representing the (fairly precise) time at which the program was compiled.

Its value is `"@2021-02-01T13:50:35.579369-05:00"`

## 8.28 Tootsville::**\*Config-File\***

### 8.28.1 Variable

\*Config-File\* names a variable:

Metadata about the configuration file last loaded

Its value is NIL

## 8.29 Tootsville::**\*Db\***

### 8.29.1 Variable

**\*Db\*** names a variable:

The default database moniker

Its value is :FRIENDLY

## 8.30 Tootsville::**\*Dbi-Connection\***

### 8.30.1 Variable

**\*Dbi-Connection\*** names a variable:

The connection selected by a WITH-MARIA block

Its value is `:NOT-CONNECTED`

## 8.31 Tootsville::**\*Elevation-Map\***

### 8.31.1 Variable

**\*Elevation-Map\*** names a variable:

The Tootangan elevation map provides a logarithmic altitude map of the approximate/net altitude of each 200 by 200 meter area of the game.

Its value is of type PNGLOAD:PNG



## 8.32 Tootsville::**\*Endpoint-List\***

### 8.32.1 Variable

**\*Endpoint-List\*** names a variable:

A list version of **\*ENDPOINTS\*** that is sometimes preferable. Both should be updated together.

Its value is of type CONS

## 8.33 Tootsville::**\*Endpoints\***

### 8.33.1 Variable

**\*Endpoints\*** names a variable:

The hash-table of all endpoints currently defined.

There is also a list version **\*ENDPOINT-LIST\*** which is preferred in some cases. Both should be updated together.

Its value is the hash table:

```

2775501260261425259
    #<ENDPOINT GET /index/html ↦ TEXT/HTML ← ENDPOINT-GET-/↦html>
2613718390945611112
    #<ENDPOINT GET / ↦ TEXT/HTML ← ENDPOINT-GET-/↦html>
4301435484693218292
    #<ENDPOINT GET /favicon/png ↦ IMAGE/PNG ← ENDPOINT-GET-
    /favicon↦png>
3605103946389453302
    #<ENDPOINT GET /favicon ↦ IMAGE/PNG ← ENDPOINT-GET-/favicon↦png>
1654812538496282845
    #<ENDPOINT GET /favicon/ico ↦ IMAGE/VND.MICROSOFT.ICON ←
    ENDPOINT-GET-/favicon/ico↦vnd.microsoft.icon>
4430847145339061514
    #<ENDPOINT GET /favicon/gif ↦ IMAGE/GIF ← ENDPOINT-GET-
    /favicon↦gif>
3661819224652773325
    #<ENDPOINT GET /favicon ↦ IMAGE/GIF ← ENDPOINT-GET-/favicon↦gif>
2357254137101586602
    #<ENDPOINT POST /login/child/json ↦ APPLICATION/JSON ←
    ENDPOINT-POST-/login/child↦json>
3079094923609898074
    #<ENDPOINT POST /login/child ↦ APPLICATION/JSON ← ENDPOINT-POST-
    /login/child↦json>
1311744403956848369
    #<ENDPOINT GET /version/about/json ↦ APPLICATION/JSON ←
    ENDPOINT-GET-/version/about↦json>
438046192794699620
    #<ENDPOINT GET /version/about ↦ APPLICATION/JSON ← ENDPOINT-GET-
    /version/about↦json>
2862236114450198699
    #<ENDPOINT GET /version/about/txt ↦ TEXT/PLAIN ← ENDPOINT-GET-
    /version/about↦txt>

```

```
4218892956418160273
  #<ENDPOINT GET /version/about ↦ TEXT/PLAIN ← ENDPOINT-GET-
  /version/about↦txt>

1455774413677645218
  #<ENDPOINT GET /version/about/detail/:PARAM/txt ↦ TEXT/PLAIN ←
  ENDPOINT-GET-/version/about/detail/param↦txt>

3568759887225338765
  #<ENDPOINT GET /version/about/detail/:PARAM ↦ TEXT/PLAIN ←
  ENDPOINT-GET-/version/about/detail/param↦txt>

693282415722877999
  #<ENDPOINT GET /version/about/detail/:PARAM/json ↦ APPLICATION/JSON
  ← ENDPOINT-GET-/version/about/detail/param↦json>

1685448398244526369
  #<ENDPOINT GET /version/about/detail/:PARAM ↦ APPLICATION/JSON ←
  ENDPOINT-GET-/version/about/detail/param↦json>

510088238736223385
  #<ENDPOINT GET /maintenance/txt ↦ TEXT/PLAIN ← ENDPOINT-GET-
  /maintenance/↦txt>

2613165918543518378
  #<ENDPOINT GET /maintenance ↦ TEXT/PLAIN ← ENDPOINT-GET-
  /maintenance/↦txt>

185362433163658785
  #<ENDPOINT POST /maintenance/quicklisp-update ↦ NIL ← ENDPOINT-
  POST-/maintenance/quicklisp-update↦nil>

1209366185538690670
  #<ENDPOINT POST /maintenance/hot-reload ↦ NIL ← ENDPOINT-POST-
  /maintenance/hot-reload↦nil>

3006916151956666734
  #<ENDPOINT POST /maintenance/buildapp ↦ NIL ← ENDPOINT-POST-
  /maintenance/buildapp↦nil>

4353388280022639758
  #<ENDPOINT POST /maintenance/buildapp/status ↦ NIL ←
  ENDPOINT-POST-/maintenance/buildapp/status↦nil>

4096376851770060843
  #<ENDPOINT POST /maintenance/reload-jscl ↦ NIL ← ENDPOINT-POST-
  /maintenance/reload-jscl↦nil>

2295846296688743859
  #<ENDPOINT POST /maintenance/quit ↦ NIL ← ENDPOINT-POST-
  /maintenance/quit↦nil>

3926858977675076860
  #<ENDPOINT GET /meta-game/services/html ↦ TEXT/HTML ← ENDPOINT-
  GET-/meta-game/services↦html>
```

4288391570599918474

#<ENDPOINT GET /meta-game/services ↪ TEXT/HTML ← ENDPOINT-GET-  
/meta-game/services↪html>

1622947186767220720

#<ENDPOINT GET /meta-game/services/old/json ↪ APPLICATION/JSON ←  
ENDPOINT-GET-/meta-game/services/old↪json>

234726932288202010

#<ENDPOINT GET /meta-game/services/old ↪ APPLICATION/JSON ←  
ENDPOINT-GET-/meta-game/services/old↪json>

2534461905518247006

#<ENDPOINT GET /meta-game/services/json ↪ APPLICATION/VND.OAI.OPENAPI;VERSION=3.0  
← ENDPOINT-GET-/meta-game/services↪json>

21519206609707318

#<ENDPOINT GET /meta-game/services ↪ APPLICATION/VND.OAI.OPENAPI;VERSION=3.0  
← ENDPOINT-GET-/meta-game/services↪json>

500367205855751383

#<ENDPOINT GET /meta-game/headers/json ↪ APPLICATION/JSON ←  
ENDPOINT-GET-/meta-game/headers↪json>

3544118970374603318

#<ENDPOINT GET /meta-game/headers ↪ APPLICATION/JSON ← ENDPOINT-  
GET-/meta-game/headers↪json>

132351661880430483

#<ENDPOINT GET /meta-game/ping/txt ↪ TEXT/PLAIN ← ENDPOINT-GET-  
/meta-game/ping↪txt>

2312565822381053381

#<ENDPOINT GET /meta-game/ping ↪ TEXT/PLAIN ← ENDPOINT-GET-/meta-  
game/ping↪txt>

3599886103547651904

#<ENDPOINT GET /gossip/ice-servers/json ↪ APPLICATION/JSON ←  
ENDPOINT-GET-/gossip/ice-servers↪json>

572200990282065073

#<ENDPOINT GET /gossip/ice-servers ↪ APPLICATION/JSON ←  
ENDPOINT-GET-/gossip/ice-servers↪json>

3464858480863409755

#<ENDPOINT POST /gossip/offers ↪ APPLICATION/SDP ← ENDPOINT-POST-  
/gossip/offers↪sdp>

3292734651114778962

#<ENDPOINT GET /gossip/offers/json ↪ APPLICATION/JSON ←  
ENDPOINT-GET-/gossip/offers↪json>

4579933893106172794

#<ENDPOINT GET /gossip/offers ↪ APPLICATION/JSON ← ENDPOINT-GET-  
/gossip/offers↪json>

```
1100913697774676897
    #<ENDPOINT POST /gossip/answers/:UUID ↦ APPLICATION/SDP ←
    ENDPOINT-POST-/gossip/answers/uuid↦sdp>
2126131175871649162
    #<ENDPOINT GET /gossip/answers/:UUID ↦ APPLICATION/SDP ←
    ENDPOINT-GET-/gossip/answers/uuid↦sdp>
734167368411805897
    #<ENDPOINT GET /toots/:TOOT-NAME/txt ↦ TEXT/PLAIN ← ENDPOINT-GET-
    /toots/toot-name↦txt>
397873300273122050
    #<ENDPOINT GET /toots/:TOOT-NAME ↦ TEXT/PLAIN ← ENDPOINT-GET-
    /toots/toot-name↦txt>
652362187253295585
    #<ENDPOINT GET /toots/:TOOT-NAME/json ↦ APPLICATION/JSON ←
    ENDPOINT-GET-/toots/toot-name↦json>
3588003849203370960
    #<ENDPOINT GET /toots/:TOOT-NAME ↦ APPLICATION/JSON ← ENDPOINT-
    GET-/toots/toot-name↦json>
798770774760887233
    #<ENDPOINT PUT /toots/:TOOT-NAME/json ↦ APPLICATION/JSON ←
    ENDPOINT-PUT-/toots/toot-name↦json>
902774860067240563
    #<ENDPOINT PUT /toots/:TOOT-NAME ↦ APPLICATION/JSON ← ENDPOINT-
    PUT-/toots/toot-name↦json>
4170731574305303361
    #<ENDPOINT POST /toots/json ↦ APPLICATION/JSON ← ENDPOINT-POST-
    /toots↦json>
1782761286852252605
    #<ENDPOINT POST /toots ↦ APPLICATION/JSON ← ENDPOINT-POST-
    /toots↦json>
2347653146422357951
    #<ENDPOINT GET /users/me/txt ↦ TEXT/PLAIN ← ENDPOINT-GET-
    /users/me↦txt>
817493754216035154
    #<ENDPOINT GET /users/me ↦ TEXT/PLAIN ← ENDPOINT-GET-
    /users/me↦txt>
911840963843854747
    #<ENDPOINT GET /users/me/json ↦ APPLICATION/JSON ← ENDPOINT-GET-
    /users/me↦json>
4131534054498028735
    #<ENDPOINT GET /users/me ↦ APPLICATION/JSON ← ENDPOINT-GET-
    /users/me↦json>
```

```
4173820247824470588
  #<ENDPOINT PUT /users/me/json ↦ APPLICATION/JSON ← ENDPOINT-PUT-
  /users/me↦json>

405515342609891492
  #<ENDPOINT PUT /users/me ↦ APPLICATION/JSON ← ENDPOINT-PUT-
  /users/me↦json>

4253363210825773152
  #<ENDPOINT PATCH /users/me/json ↦ APPLICATION/JSON ←
  ENDPOINT-PATCH-/users/me↦json>

3556788904925915638
  #<ENDPOINT PATCH /users/me ↦ APPLICATION/JSON ← ENDPOINT-PATCH-
  /users/me↦json>

1873952710285345362
  #<ENDPOINT GET /users/me/toots/txt ↦ TEXT/PLAIN ← ENDPOINT-GET-
  /users/me/toots↦txt>

4001463141395061250
  #<ENDPOINT GET /users/me/toots ↦ TEXT/PLAIN ← ENDPOINT-GET-
  /users/me/toots↦txt>

2200278265671935120
  #<ENDPOINT GET /users/me/toots/json ↦ APPLICATION/JSON ←
  ENDPOINT-GET-/users/me/toots↦json>

800366382555165045
  #<ENDPOINT GET /users/me/toots ↦ APPLICATION/JSON ← ENDPOINT-GET-
  /users/me/toots↦json>

2125156083082933535
  #<ENDPOINT GET /users/me/toots/:TOOT-NAME/txt ↦ TEXT/PLAIN ←
  ENDPOINT-GET-/users/me/toots/toot-name↦txt>

3159196718401479487
  #<ENDPOINT GET /users/me/toots/:TOOT-NAME ↦ TEXT/PLAIN ←
  ENDPOINT-GET-/users/me/toots/toot-name↦txt>

1070605157160274909
  #<ENDPOINT GET /users/me/toots/:TOOT-NAME/json ↦ APPLICATION/JSON
  ← ENDPOINT-GET-/users/me/toots/toot-name↦json>

3220404487643901754
  #<ENDPOINT GET /users/me/toots/:TOOT-NAME ↦ APPLICATION/JSON ←
  ENDPOINT-GET-/users/me/toots/toot-name↦json>

4025206783682213707
  #<ENDPOINT DELETE /users/me/toots/:TOOT-NAME/json ↦ APPLICATION/JSON
  ← ENDPOINT-DELETE-/users/me/toots/toot-name↦json>

164843851602781227
  #<ENDPOINT DELETE /users/me/toots/:TOOT-NAME ↦ APPLICATION/JSON ←
  ENDPOINT-DELETE-/users/me/toots/toot-name↦json>
```

3308243433724706606

```
#<ENDPOINT POST /users/me/play-with/:TOOT-NAME/json ↳  
APPLICATION/JSON ← ENDPOINT-POST-/users/me/play-with/toot-  
name↳json>
```

635025581045997038

```
#<ENDPOINT POST /users/me/play-with/:TOOT-NAME ↳ APPLICATION/JSON  
← ENDPOINT-POST-/users/me/play-with/toot-name↳json>
```

2725175015971607415

```
#<ENDPOINT GET /world/json ↳ APPLICATION/JSON ← ENDPOINT-GET-  
/world↳json>
```

3271757561095108054

```
#<ENDPOINT GET /world ↳ APPLICATION/JSON ← ENDPOINT-GET-  
/world↳json>
```

385524676063845658

```
#<ENDPOINT GET /world/tootanga/:LATITUDE/:LONGITUDE/:ALTITUDE/json  
↳ APPLICATION/JSON ← ENDPOINT-GET-/world/tootanga/latitude/longitude/altitude↳j
```

3423369905937201739

```
#<ENDPOINT GET /world/tootanga/:LATITUDE/:LONGITUDE/:ALTITUDE ↳  
APPLICATION/JSON ← ENDPOINT-GET-/world/tootanga/latitude/longitude/altitude↳json
```

3010244976934717594

```
#<ENDPOINT GET /world/clock/date/txt ↳ TEXT/PLAIN ← ENDPOINT-GET-  
/world/clock/date↳txt>
```

3542956409881927519

```
#<ENDPOINT GET /world/clock/date ↳ TEXT/PLAIN ← ENDPOINT-GET-  
/world/clock/date↳txt>
```

903662128716469545

```
#<ENDPOINT GET /world/clock/date/long/txt ↳ TEXT/PLAIN ←  
ENDPOINT-GET-/world/clock/date/long↳txt>
```

4521263252416595045

```
#<ENDPOINT GET /world/clock/date/long ↳ TEXT/PLAIN ←  
ENDPOINT-GET-/world/clock/date/long↳txt>
```

4474702625315267486

```
#<ENDPOINT GET /world/clock/date/abbrev/txt ↳ TEXT/PLAIN ←  
ENDPOINT-GET-/world/clock/date/abbrev↳txt>
```

4044962940318310263

```
#<ENDPOINT GET /world/clock/date/abbrev ↳ TEXT/PLAIN ←  
ENDPOINT-GET-/world/clock/date/abbrev↳txt>
```

1234443831199153308

```
#<ENDPOINT GET /world/clock/time/json ↳ APPLICATION/JSON ←  
ENDPOINT-GET-/world/clock/time↳json>
```

64981768924257163

```
#<ENDPOINT GET /world/clock/time ↦ APPLICATION/JSON ←
ENDPOINT-GET-/world/clock/time↦json>
```

3128562978297264469

```
#<ENDPOINT GET /world/clock/time/txt ↦ TEXT/PLAIN ← ENDPOINT-GET-
/world/clock/time↦txt>
```

1220359457604711210

```
#<ENDPOINT GET /world/clock/time ↦ TEXT/PLAIN ← ENDPOINT-GET-
/world/clock/time↦txt>
```

474134724174379242

```
#<ENDPOINT GET /world/clock/calendar/year/:YEAR/month/:MONTH/fragment/html
↦ TEXT/HTML ← ENDPOINT-GET-/world/clock/calendar/year/year/month/month/fragment-
```

4465150788489470284

```
#<ENDPOINT GET /world/clock/calendar/year/:YEAR/month/:MONTH/fragment
↦ TEXT/HTML ← ENDPOINT-GET-/world/clock/calendar/year/year/month/month/fragment-
```

1995753829732542241

```
#<ENDPOINT GET /world/clock/calendar/now/fragment/html ↦
TEXT/HTML ← ENDPOINT-GET-/world/clock/calendar/now/fragment↦html>
```

990868233327192808

```
#<ENDPOINT GET /world/clock/calendar/now/fragment ↦ TEXT/HTML ←
ENDPOINT-GET-/world/clock/calendar/now/fragment↦html>
```

3830107823585105774

```
#<ENDPOINT GET /world/clock/calendar/year/:YEAR/fragment/html ↦
TEXT/HTML ← ENDPOINT-GET-/world/clock/calendar/year/year/fragment↦html>
```

2898106294566400561

```
#<ENDPOINT GET /world/clock/calendar/year/:YEAR/fragment ↦
TEXT/HTML ← ENDPOINT-GET-/world/clock/calendar/year/year/fragment↦html>
```

1441330249852351824

```
#<ENDPOINT GET /world/clock/calendar/year/:YEAR/month/:MONTH/html
↦ TEXT/HTML ← ENDPOINT-GET-/world/clock/calendar/year/year/month/month↦html>
```

3107319515624852900

```
#<ENDPOINT GET /world/clock/calendar/year/:YEAR/month/:MONTH ↦
TEXT/HTML ← ENDPOINT-GET-/world/clock/calendar/year/year/month/month↦html>
```

4050907988209917166

```
#<ENDPOINT GET /world/clock/time/detailed/txt ↦ TEXT/PLAIN ←
ENDPOINT-GET-/world/clock/time/detailed↦txt>
```

3710653948837694751

```
#<ENDPOINT GET /world/clock/time/detailed ↦ TEXT/PLAIN ←
ENDPOINT-GET-/world/clock/time/detailed↦txt>
```

3796390286142599632

```
#<ENDPOINT GET /world/sky/tootanga/:LATITUDE/:LONGITUDE/json ↦
APPLICATION/JSON ← ENDPOINT-GET-/world/sky/tootanga/latitude/longitude↦json>
```



```
2444360649006507045
    #<ENDPOINT GET /world/sky/tootanga/:LATITUDE/:LONGITUDE ↦
    APPLICATION/JSON ← ENDPOINT-GET-/world/sky/tootanga/latitude/longitude↦json>■

3899839684376630916
    #<ENDPOINT POST /gossip/twilio/incoming/call ↦ TEXT/XML ←
    ENDPOINT-POST-/gossip/twilio/incoming/call↦xml>

14645949841099209
    #<ENDPOINT POST /gossip/twilio/incoming/fax ↦ TEXT/XML ←
    ENDPOINT-POST-/gossip/twilio/incoming/fax↦xml>

1490427615918773705
    #<ENDPOINT POST /gossip/twilio/incoming/sms ↦ TEXT/XML ←
    ENDPOINT-POST-/gossip/twilio/incoming/sms↦xml>

1702692747509867472
    #<ENDPOINT POST /gossip/twilio/incoming/whatsapp ↦ TEXT/XML ←
    ENDPOINT-POST-/gossip/twilio/incoming/whatsapp↦xml>

2248636144567731984
    #<ENDPOINT POST /gossip/twilio/incoming/verify ↦ TEXT/XML ←
    ENDPOINT-POST-/gossip/twilio/incoming/verify↦xml>

2538184196477881734
    #<ENDPOINT POST /gossip/alexa/info/region/:REGION/json ↦
    APPLICATION/JSON ← ENDPOINT-POST-/gossip/alexa/info/region/region↦json>■

4451440760646442058
    #<ENDPOINT POST /gossip/alexa/info/region/:REGION ↦ APPLICATION/JSON■
    ← ENDPOINT-POST-/gossip/alexa/info/region/region↦json>

3129363612469868778
    #<ENDPOINT POST /gossip/alexa/chat/region/:REGION/json ↦
    APPLICATION/JSON ← ENDPOINT-POST-/gossip/alexa/chat/region/region↦json>■

3772889472112935846
    #<ENDPOINT POST /gossip/alexa/chat/region/:REGION ↦ APPLICATION/JSON■
    ← ENDPOINT-POST-/gossip/alexa/chat/region/region↦json>

2914246148950682845
    #<ENDPOINT POST /gossip/alexa/clock/region/:REGION/json ↦
    APPLICATION/JSON ← ENDPOINT-POST-/gossip/alexa/clock/region/region↦json>■

3607211951621492359
    #<ENDPOINT POST /gossip/alexa/clock/region/:REGION ↦
    APPLICATION/JSON ← ENDPOINT-POST-/gossip/alexa/clock/region/region↦json>■

3514820420918378876
    #<ENDPOINT POST /world/infinity/json ↦ APPLICATION/JSON ←
    ENDPOINT-POST-/world/infinity↦json>

2983796640953129098
    #<ENDPOINT POST /world/infinity ↦ APPLICATION/JSON ←
    ENDPOINT-POST-/world/infinity↦json>
```

4297781951900874698

```
#<ENDPOINT POST /world/infinity/add-furniture/json ↦  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/add-  
furniture↦json>
```

4153708598936678059

```
#<ENDPOINT POST /world/infinity/add-furniture ↦ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/add-furniture↦json>
```

3534633058848260767

```
#<ENDPOINT POST /world/infinity/add-to-list/json ↦ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/add-to-list↦json>
```

3160128725037955181

```
#<ENDPOINT POST /world/infinity/add-to-list ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/add-to-list↦json>
```

886494176412242955

```
#<ENDPOINT POST /world/infinity/click/json ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/click↦json>
```

4605344589893295780

```
#<ENDPOINT POST /world/infinity/click ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/click↦json>
```

2769295429646498886

```
#<ENDPOINT POST /world/infinity/create-user-house/json ↦  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/create-user-  
house↦json>
```

3156673351385279550

```
#<ENDPOINT POST /world/infinity/create-user-house ↦ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/create-user-house↦json>
```

2034904767075969688

```
#<ENDPOINT POST /world/infinity/dofff/json ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/dofff↦json>
```

2856840100888173339

```
#<ENDPOINT POST /world/infinity/dofff ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/dofff↦json>
```

2115080225378518581

```
#<ENDPOINT POST /world/infinity/don/json ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/don↦json>
```

86474508769345563

```
#<ENDPOINT POST /world/infinity/don ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/don↦json>
```

600502829093376564

```
#<ENDPOINT POST /world/infinity/echo/json ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/echo↦json>
```

```
1407229498174929943
  #<ENDPOINT POST /world/infinity/echo ↳ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/echo↳json>

1818181793002447105
  #<ENDPOINT POST /world/infinity/finger/json ↳ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/finger↳json>

411075493542412538
  #<ENDPOINT POST /world/infinity/finger ↳ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/finger↳json>

1426436054577212618
  #<ENDPOINT POST /world/infinity/game-action/json ↳ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/game-action↳json>

2905404326129432418
  #<ENDPOINT POST /world/infinity/game-action ↳ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/game-action↳json>

3612501823886010390
  #<ENDPOINT POST /world/infinity/get-avatars/json ↳ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/get-avatars↳json>

90288663164151526
  #<ENDPOINT POST /world/infinity/get-avatars ↳ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/get-avatars↳json>

1296999664440820690
  #<ENDPOINT POST /world/infinity/get-color-palettes/json ↳
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-color-
  palettes↳json>

4027751240528359165
  #<ENDPOINT POST /world/infinity/get-color-palettes ↳
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-color-
  palettes↳json>

4532850485041467476
  #<ENDPOINT POST /world/infinity/get-inventory/json ↳
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-
  inventory↳json>

243021057524799367
  #<ENDPOINT POST /world/infinity/get-inventory ↳ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/get-inventory↳json>

4167155213974489996
  #<ENDPOINT POST /world/infinity/get-inventory-by-type/json ↳
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-inventory-
  by-type↳json>
```

1047054423369345559

```
#<ENDPOINT POST /world/infinity/get-inventory-by-type ↪  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-inventory-  
by-type↪json>
```

815127266796768811

```
#<ENDPOINT POST /world/infinity/get-online-users/json ↪  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-online-  
users↪json>
```

4106792282579579131

```
#<ENDPOINT POST /world/infinity/get-online-users ↪ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/get-online-users↪json>
```

1530669576519352676

```
#<ENDPOINT POST /world/infinity/get-room-list/json ↪  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-room-  
list↪json>
```

1229240807035145934

```
#<ENDPOINT POST /world/infinity/get-room-list ↪ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/get-room-list↪json>
```

3632934770832966793

```
#<ENDPOINT POST /world/infinity/get-server-time/json ↪  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-server-  
time↪json>
```

723702752102433611

```
#<ENDPOINT POST /world/infinity/get-server-time ↪ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/get-server-time↪json>
```

2381679612391285307

```
#<ENDPOINT POST /world/infinity/get-session-apple/json ↪  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-session-  
apple↪json>
```

1941036288412020259

```
#<ENDPOINT POST /world/infinity/get-session-apple ↪ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/get-session-apple↪json>
```

2545589777087219548

```
#<ENDPOINT POST /world/infinity/get-store-item-info/json ↪  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-store-item-  
info↪json>
```

3591109234948706272

```
#<ENDPOINT POST /world/infinity/get-store-item-info ↪  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-store-  
item-info↪json>
```

```
388236602821079676
  #<ENDPOINT POST /world/infinity/get-user-lists/json ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-user-
  lists↦json>

4026062155054880834
  #<ENDPOINT POST /world/infinity/get-user-lists ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/get-user-lists↦json>

1867351472563902820
  #<ENDPOINT POST /world/infinity/get-wallet/json ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/get-wallet↦json>

743393508375587111
  #<ENDPOINT POST /world/infinity/get-wallet ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/get-wallet↦json>

1487464466660955791
  #<ENDPOINT POST /world/infinity/get-zone-list/json ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-zone-
  list↦json>

3499294289065798086
  #<ENDPOINT POST /world/infinity/get-zone-list ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/get-zone-list↦json>

1742512978231992899
  #<ENDPOINT POST /world/infinity/give/json ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/give↦json>

3186022246400269368
  #<ENDPOINT POST /world/infinity/give ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/give↦json>

3393910229766804491
  #<ENDPOINT POST /world/infinity/go/json ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/go↦json>

2345155363327254156
  #<ENDPOINT POST /world/infinity/go ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/go↦json>

111246169399534578
  #<ENDPOINT POST /world/infinity/init-user-room/json ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/init-user-
  room↦json>

3645887930300868921
  #<ENDPOINT POST /world/infinity/init-user-room ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/init-user-room↦json>

2246548973544009574
  #<ENDPOINT POST /world/infinity/join/json ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/join↦json>
```

```
4579938525002377968
  #<ENDPOINT POST /world/infinity/join ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/join↦json>

1434158854020747068
  #<ENDPOINT POST /world/infinity/logout/json ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/logout↦json>

3302013364249440874
  #<ENDPOINT POST /world/infinity/logout ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/logout↦json>

4237490832679781235
  #<ENDPOINT POST /world/infinity/mail-customer-service/json ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/mail-customer-
  service↦json>

2053750045881607118
  #<ENDPOINT POST /world/infinity/mail-customer-service ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/mail-customer-
  service↦json>

3279637421459493870
  #<ENDPOINT POST /world/infinity/peek-at-inventory/json ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/peek-at-
  inventory↦json>

31126383443591073
  #<ENDPOINT POST /world/infinity/peek-at-inventory ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/peek-at-inventory↦json>

2114634252262689739
  #<ENDPOINT POST /world/infinity/ping/json ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/ping↦json>

1497657590906864494
  #<ENDPOINT POST /world/infinity/ping ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/ping↦json>

3944391893719435316
  #<ENDPOINT POST /world/infinity/prompt-reply/json ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/prompt-reply↦json>

3447085172245388639
  #<ENDPOINT POST /world/infinity/prompt-reply ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/prompt-reply↦json>

4275937288897563148
  #<ENDPOINT POST /world/infinity/remove-from-list/json ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/remove-from-
  list↦json>

4399652790573300362
  #<ENDPOINT POST /world/infinity/remove-from-list ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/remove-from-list↦json>
```

```
715734164857350997
  #<ENDPOINT POST /world/infinity/report-bug/json ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/report-bug↦json>

1142409125308217021
  #<ENDPOINT POST /world/infinity/report-bug ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/report-bug↦json>

2929010456783785717
  #<ENDPOINT POST /world/infinity/report-user/json ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/report-user↦json>

4347007719556469473
  #<ENDPOINT POST /world/infinity/report-user ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/report-user↦json>

274901334464311133
  #<ENDPOINT POST /world/infinity/request-buddy/json ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/request-
  buddy↦json>

3758851943699585440
  #<ENDPOINT POST /world/infinity/request-buddy ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/request-buddy↦json>

1463511803243320041
  #<ENDPOINT POST /world/infinity/send-out-of-band-message/json ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/send-out-of-
  band-message↦json>

2353418606147398
  #<ENDPOINT POST /world/infinity/send-out-of-band-message ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/send-out-of-
  band-message↦json>

3630072932184348066
  #<ENDPOINT POST /world/infinity/server-time/json ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/server-time↦json>

3995549550556215699
  #<ENDPOINT POST /world/infinity/server-time ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/server-time↦json>

3612183101183848272
  #<ENDPOINT POST /world/infinity/set-avatar-color/json ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/set-avatar-
  color↦json>

2261376797728342278
  #<ENDPOINT POST /world/infinity/set-avatar-color ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/set-avatar-color↦json>
```

3252339162700958257

```
#<ENDPOINT POST /world/infinity/set-furniture/json ↦  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/set-  
furniture↦json>
```

162550141888702464

```
#<ENDPOINT POST /world/infinity/set-furniture ↦ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/set-furniture↦json>
```

1208366953458090133

```
#<ENDPOINT POST /world/infinity/set-room-var/json ↦ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/set-room-var↦json>
```

4445291985524611193

```
#<ENDPOINT POST /world/infinity/set-room-var ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/set-room-var↦json>
```

3156243483016127915

```
#<ENDPOINT POST /world/infinity/set-user-var/json ↦ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/set-user-var↦json>
```

3788583612421438164

```
#<ENDPOINT POST /world/infinity/set-user-var ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/set-user-var↦json>
```

3714301876461587212

```
#<ENDPOINT POST /world/infinity/spawn-zone/json ↦ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/spawn-zone↦json>
```

677933659063334801

```
#<ENDPOINT POST /world/infinity/spawn-zone ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/spawn-zone↦json>
```

4574057132262960364

```
#<ENDPOINT POST /world/infinity/speak/json ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/speak↦json>
```

2537548613447673980

```
#<ENDPOINT POST /world/infinity/speak ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/speak↦json>
```

3625523948320548380

```
#<ENDPOINT POST /world/infinity/start-event/json ↦ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/start-event↦json>
```

28145633782456136

```
#<ENDPOINT POST /world/infinity/start-event ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/start-event↦json>
```

3573642821779180528

```
#<ENDPOINT POST /world/infinity/end-event/json ↦ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/end-event↦json>
```



```
3971243235013808867
  #<ENDPOINT POST /world/infinity/end-event ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/end-event↦json>

4094027678187318451
  #<ENDPOINT POST /world/infinity/use-equipment/json ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/use-
  equipment↦json>

2000081458093256992
  #<ENDPOINT POST /world/infinity/use-equipment ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/use-equipment↦json>

4204947670449303791
  #<ENDPOINT POST /world/infinity/add-journal-entry/json ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/add-journal-
  entry↦json>

3892582786088941557
  #<ENDPOINT POST /world/infinity/add-journal-entry ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/add-journal-entry↦json>

1375612565279927928
  #<ENDPOINT POST /world/infinity/delete-mail-message/json ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/delete-mail-
  message↦json>

4436290906837033172
  #<ENDPOINT POST /world/infinity/delete-mail-message ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/delete-
  mail-message↦json>

2230697178133601397
  #<ENDPOINT POST /world/infinity/doff/json ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/doff↦json>

2262727635859522983
  #<ENDPOINT POST /world/infinity/doff ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/doff↦json>

2941269994842038810
  #<ENDPOINT POST /world/infinity/get-mail-in-box/json ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-mail-
  in-box↦json>

1747771810566703308
  #<ENDPOINT POST /world/infinity/get-mail-in-box ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/get-mail-in-box↦json>

3798885911312028850
  #<ENDPOINT POST /world/infinity/get-passport/json ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/get-passport↦json>
```

2904598588733088666  
#<ENDPOINT POST /world/infinity/get-passport ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/get-passport↦json>

2939097456470116256  
#<ENDPOINT POST /world/infinity/send-mail-message/json ↦  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/send-mail-  
message↦json>

2969945001075424066  
#<ENDPOINT POST /world/infinity/send-mail-message ↦ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/send-mail-message↦json>

1779739157844399139  
#<ENDPOINT POST /world/infinity/stamp-passport/json ↦  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/stamp-  
passport↦json>

3439491491755883883  
#<ENDPOINT POST /world/infinity/stamp-passport ↦ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/stamp-passport↦json>

3292086240137672046  
#<ENDPOINT POST /world/infinity/enumerate-wear-slots/json ↦  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/enumerate-wear-  
slots↦json>

2976415471988222329  
#<ENDPOINT POST /world/infinity/enumerate-wear-slots ↦  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/enumerate-  
wear-slots↦json>

3910763075184512485  
#<ENDPOINT POST /world/infinity/wardrobe/json ↦ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/wardrobe↦json>

1812566037956311588  
#<ENDPOINT POST /world/infinity/wardrobe ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/wardrobe↦json>

1632442791808375756  
#<ENDPOINT POST /world/infinity/get-room-vars/json ↦  
APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/get-room-  
vars↦json>

468288789791614400  
#<ENDPOINT POST /world/infinity/get-room-vars ↦ APPLICATION/JSON  
← ENDPOINT-POST-/world/infinity/get-room-vars↦json>

1675699997633227251  
#<ENDPOINT POST /world/infinity/wtl/json ↦ APPLICATION/JSON ←  
ENDPOINT-POST-/world/infinity/wtl↦json>

```
935878708885882574
  #<ENDPOINT POST /world/infinity/wtl ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/wtl↦json>

152427975052322824
  #<ENDPOINT POST /world/infinity/wtl-4/json ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/wtl-4↦json>

4400131562262490604
  #<ENDPOINT POST /world/infinity/wtl-4 ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/wtl-4↦json>

4425915036243019463
  #<ENDPOINT POST /world/infinity/shoot/json ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/shoot↦json>

2806337527219004503
  #<ENDPOINT POST /world/infinity/shoot ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/shoot↦json>

1230518630342143647
  #<ENDPOINT POST /world/infinity/toot-list/json ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/toot-list↦json>

4335180912172648585
  #<ENDPOINT POST /world/infinity/toot-list ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/toot-list↦json>

2404520522177971559
  #<ENDPOINT POST /world/infinity/play-with/json ↦ APPLICATION/JSON
  ← ENDPOINT-POST-/world/infinity/play-with↦json>

211099490374444857
  #<ENDPOINT POST /world/infinity/play-with ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/play-with↦json>

949834039714802218
  #<ENDPOINT POST /world/infinity/quiesce/json ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/quiesce↦json>

4538439985630186268
  #<ENDPOINT POST /world/infinity/quiesce ↦ APPLICATION/JSON ←
  ENDPOINT-POST-/world/infinity/quiesce↦json>

4104464769970938872
  #<ENDPOINT POST /world/infinity/consider-child-approval/json ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/consider-child-
  approval↦json>

1630414277483219540
  #<ENDPOINT POST /world/infinity/consider-child-approval ↦
  APPLICATION/JSON ← ENDPOINT-POST-/world/infinity/consider-child-
  approval↦json>
```

3222404907486997097

#<ENDPOINT POST /world/infinity/read-map/json ↦ APPLICATION/JSON  
<- ENDPOINT-POST-/world/infinity/read-map↦json>

4162667056903443726

#<ENDPOINT POST /world/infinity/read-map ↦ APPLICATION/JSON <-  
ENDPOINT-POST-/world/infinity/read-map↦json>

## 8.34 Tootsville::**\*Extensions-For-Content-Types\***

### 8.34.1 Variable

**\*Extensions-For-Content-Types\*** names an undocumented variable with the value of type CONS

## 8.35 Tootsville::**\*Google-Account-Keys-Refresh\***

### 8.35.1 Variable

\*Google-Account-Keys-Refresh\* names a variable:

How often (in sec) to refresh the Google account keys?

These are used in Firebase authentication verification, but only in the event Cache-Control: max-age is not set on the keys, which it usually is.

Its value is 1,200 (#x4B0)

## 8.36 Tootsville::**\*Habitat-Map\***

### 8.36.1 Variable

**\*Habitat-Map\*** names a variable:

The Tootanga map contains color-coded pixels representing the various habitat areas of the game. Each pixel represents a 200m by 200m area; thus, the entire map area (800 by 600 pixels) represents a playable game area of 160 by 120 km.

Its value is of type PNGLOAD:PNG

## 8.37 Tootsville::**\*Http-Status-Message\***

### 8.37.1 Variable

**\*Http-Status-Message\*** names an undocumented variable with the value the hash table:

100 Continue, please. I'd like to hear more.

101 Switching Protocols

200 Okie-dokie, here you go!

201 Look at what I've made now

202 I'll take that, sure

203 I'm not really sure, but ...

204 Here's the nothing you wanted

205 Reset Content

206 Partial Content

207 Multi-Status

300 Multiple Choices

301 Moved Permanently

302 Moved Temporarily

303 See Other

304 Not Modified

305 Use Proxy

307 Temporary Redirect

400 Bad Request

401 Authorization Required

402 Payment Required

403 Forbidden

404 Not Found

405 Method Not Allowed

406 Not Acceptable

407 Proxy Authentication Required

408 Request Time-out

409 Conflict

410 Gone

411 Length Required



412	Precondition Failed
413	Request Entity Too Large
414	Request-URI Too Large
415	Unsupported Media Type
416	Requested range not satisfiable
417	Expectation Failed
422	Unprocessable Entity
424	Failed Dependency
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Time-out
505	Version not supported

## 8.38 Tootsville::**\*Humidity-Field\***

### 8.38.1 Variable

**\*Humidity-Field\*** names a variable:

The humidity field for the entire island of Tootanga.

Its value is of type (SIMPLE-ARRAY T (800 600))

## **8.39 Tootsville::`*Ice-Credentials*`**

### **8.39.1 Variable**

`*Ice-Credentials*` names an undocumented variable with the value `NIL`

## 8.40 Tootsville::**\*Infinity-Ops\***

### 8.40.1 Variable

**\*Infinity-Ops\*** names an undocumented variable with the value NIL

## 8.41 Tootsville::**\*Infinity-Rest-Requests\***

### 8.41.1 Variable

**\*Infinity-Rest-Requests\*** names an undocumented variable with the value 0 (`#x0`)

## 8.42 Tootsville::**\*Infinity-Stream-Requests\***

### 8.42.1 Variable

**\*Infinity-Stream-Requests\*** names an undocumented variable with the value 0 (#x0)

## 8.43 Tootsville::**\*Infinity-Users\***

### 8.43.1 Variable

**\*Infinity-Users\*** names an undocumented variable with the value the hash table:

## 8.44 Tootsville::**\*Infinity-Websocket-Resource\***

### 8.44.1 Variable

**\*Infinity-Websocket-Resource\*** names an undocumented variable with the value of type `TOOTSVILLE::INFINITY-WEB_SOCKET-RESOURCE`



## **8.45 Tootsville::~~Maintenance-Tasks-Performed~~\***

### **8.45.1 Variable**

~~Maintenance-Tasks-Performed~~\* names an undocumented variable with the value NIL

## 8.46 Tootsville::**\*Metronome\***

### 8.46.1 Variable

**\*Metronome\*** names an undocumented variable with the value NIL

## 8.47 Tootsville::**\*Metronome-Next-Tick\***

### 8.47.1 Variable

**\*Metronome-Next-Tick\*** names a variable:

The time at which the Metronome should next “tick”.

Its value is 3,821,194,235 (#xE3C2CBFB)

## 8.48 Tootsville::**\*Metronome-Run\***

### 8.48.1 Variable

**\*Metronome-Run\*** names an undocumented variable with the value T

## 8.49 Tootsville::**\*Metronome-Task-Lock\***

### 8.49.1 Variable

**\*Metronome-Task-Lock\*** names a variable:

A lock used to protect inter-thread access to the Metronome tasks.

Its value is NIL

## 8.50 Tootsville::**\*Metronome-Tasks\***

### 8.50.1 Variable

**\*Metronome-Tasks\*** names an undocumented variable with the value NIL

## 8.51 Tootsville::**\*Motd\***

### 8.51.1 Variable

**\*Motd\*** names a variable:

The message of the day.

This is served up to every person who signs in. It can be altered easily using the Section 7.56 [TOOTSVILLE-USER MOTD], page 184, command.

Its value is `"Welcome to Tootsville! Let's make some noise!`

`This is experimental server software for Tootsville V."`

## 8.52 Tootsville::**\*NPC-List\***

### 8.52.1 Variable

**\*NPC-List\*** names an undocumented variable with the value the hash table:

ZAP	ZAP
FLORA	FLORA
SUPERSTAR	SUPERSTAR
LIL-MC	LIL-MC
CUPID	CUPID
MOO	MOO
DOTTIE	DOTTIE
SPARKLE	SPARKLE
DOODLE	DOODLE
PICASSO	PICASSO
HARMONY	HARMONY
PROPS	PROPS
RAD	RAD
CHAOS	CHAOS
SMUDGE	SMUDGE
SPLOOT	SPLOOT
NEVERMIND	NEVERMIND
SHADE	SHADE
JACK	JACK
SNOWCONE	SNOWCONE
MAYOR-LOUIS	MAYOR-LOUIS



## 8.53 Tootsville::**\*Post-Tests-Queue\***

### 8.53.1 Variable

**\*Post-Tests-Queue\*** names a variable:

Power-on-self-tests are placed into this queue, usually by DEFPOST.

Its value is of type CONS

## 8.54 Tootsville::**\*Robots\***

### 8.54.1 Variable

**\*Robots\*** names a variable:

All robots currently active in the game world from this node.

Its value is the hash table:

## **8.55 Tootsville::**\*Running-Main-Loop\*****

### **8.55.1 Variable**

**\*Running-Main-Loop\*** names an undocumented variable with the value NIL

## 8.56 Tootsville::**\*Started\***

### 8.56.1 Variable

**\*Started\*** names a variable:

The time at which the server was started

Its value is NIL

## **8.57 Tootsville::**\*Tcp-Clients\*****

### **8.57.1 Variable**

**\*Tcp-Clients\*** names an undocumented variable with the value the hash table:

## 8.58 Tootsville::**\*Tcp-Listener\***

### 8.58.1 Variable

**\*Tcp-Listener\*** names an undocumented variable with the value NIL

## 8.59 Tootsville::**\*Tcp-Peer-Traffic\***

### 8.59.1 Variable

**\*Tcp-Peer-Traffic\*** names an undocumented variable with the value 0 (**#x0**)

## 8.60 Tootsville::**\*The-Metronome-Thread\***

### 8.60.1 Variable

**\*The-Metronome-Thread\*** names a variable:

The thread from which the metronome's coördination efforts are conducted.

Its value is NIL



## 8.61 Tootsville::**\*Toot\***

### 8.61.1 Variable

**\*Toot\*** names a variable:

The Toot that the active user, is currently using.

Its value is NIL

## 8.62 Tootsville::**\*Trace-Output-Heartbeat-Time\***

### 8.62.1 Variable

**\*Trace-Output-Heartbeat-Time\*** names a variable:

A thread listing is dumped every **\*TRACE-OUTPUT-HEARTBEAT-TIME\*** seconds into the verbose log.

Its value is 90 (#x5A)

## 8.63 Tootsville::**\*User\***

### 8.63.1 Variable

**\*User\*** names a variable:

The currently-signed-in user, if any

Its value is NIL

## 8.64 Tootsville::**\*Utc-Timezone\***

### 8.64.1 Variable

**\*Utc-Timezone\*** names a variable:

The UTC time zone.

The Universal Coördinated Time time zone.

For practical purposes, this is essentially the same as GMT (Greenwich Mean Time) or Z (Zulu Time).

Its value is of type LOCAL-TIME::TIMEZONE

## 8.65 Tootsville::**\*Verbose-Logging-Lock\***

### 8.65.1 Variable

**\*Verbose-Logging-Lock\*** names a variable:

A lock used to prevent the Verbose library from cross-talking over itself.

When multiple threads try to write at the same time, you can get partial messages mixed together in a confusing way. This lock prevents that from occurring when we use our definition of `VERBOSE::FORMAT-MESSAGE` (not in this manual) which observes it.

Its value is of type `SB-THREAD:MUTEX`

## **8.66 Tootsville::**\*Weather-Kernel\*****

### **8.66.1 Variable**

\*Weather-Kernel\* names an undocumented variable with the value NIL

## 8.67 Tootsville::**\*Websocket-Server\***

### 8.67.1 Variable

**\*Websocket-Server\*** names a variable:

The Hunchentoot/Hunchensocket server object for WebSockets.

Its value is NIL

## 8.68 Tootsville::**\*Wind-Vector-Field\***

### 8.68.1 Variable

**\*Wind-Vector-Field\*** names a variable:

The wind vector field for the entire island of Tootanga.

Its value is of type (SIMPLE-ARRAY T (800 600))



## 8.69 Tootsville::**\*Ws-Chars-Broadcast\***

### 8.69.1 Variable

**\*Ws-Chars-Broadcast\*** names a variable:

Total payload characters broadcast.

NB you'd have to multiply this by connected clients to get a real idea of the bandwidth involved.

Its value is 0 (#x0)

## 8.70 Tootsville::**\*Ws-Chars-Received\***

### 8.70.1 Variable

**\*Ws-Chars-Received\*** names a variable:

Total payload characters read.

Its value is 0 (#x0)

## 8.71 Tootsville::**\*Ws-Chars-Unicast\***

### 8.71.1 Variable

**\*Ws-Chars-Unicast\*** names a variable:

Total payload characters unicast to anyone.

Its value is 0 (#x0)

## 8.72 Tootsville::**\*Ws-Client-For-Toot\***

### 8.72.1 Variable

**\*Ws-Client-For-Toot\*** names an undocumented variable with the value the hash table:

## **8.73 Tootsville::**\*Ws-Client-For-User\*****

### **8.73.1 Variable**

**\*Ws-Client-For-User\*** names an undocumented variable with the value the hash table:

## 8.74 Tootsville::**\*Ws-Connections\***

### 8.74.1 Variable

**\*Ws-Connections\*** names a variable:

The number of times that someone has connected ever. *NOT* the same as *active* connections.

Its value is 0 (#x0)

## 8.75 Tootsville::**\*Ws-Sign-Ins\***

### 8.75.1 Variable

**\*Ws-Sign-Ins\*** names a variable:

The number of times that someone has authenticated (signed in) ever.

Its value is 0 (#x0)

## 8.76 Tootsville::**\*Ws-Surprise-Disconnects\***

### 8.76.1 Variable

**\*Ws-Surprise-Disconnects\*** names a variable:

Number of times someone has dropped a connection without a proper disconnection sequence.

Its value is 0 (#x0)



## 8.77 Tootsville::**\*Ws-Traffic-Commands\***

### 8.77.1 Variable

**\*Ws-Traffic-Commands\*** names an undocumented variable with the value the hash table:

## 8.78 Tootsville::**\*Ws-Traffic-From\***

### 8.78.1 Variable

**\*Ws-Traffic-From\*** names an undocumented variable with the value the hash table:

## 8.79 Tootsville::**\*Ws-Traffic-Other\***

### 8.79.1 Variable

**\*Ws-Traffic-Other\*** names an undocumented variable with the value 0 (`#x0`)

## 8.80 Tootsville::+Alexa-Timestamp-Tolerance+

### 8.80.1 Variable

+Alexa-Timestamp-Tolerance+ names a variable:

Amazon requires we request queries with a timestamp more than  $\pm$  this many seconds.

Its value is 150 (#x96)

## 8.81 Tootsville::+Amazon-Cert-Chain-Url-Matching+

### 8.81.1 Variable

+Amazon-Cert-Chain-Url-Matching+ names a variable:

list of pairs of strings and comparison functions which must be met for the URL of an Alexa certificate chain. See Section 8.190 [TOOTSVILLE CHECK-ALEXA-SIGNATURE-CERT-CHAIN-URL], page 444,

Its value is of type CONS

## 8.82 Tootsville::+Backtrace-Regex+

### 8.82.1 Variable

+Backtrace-Regex+ names a variable:

A regular expression to split backtraces

Its value is `"\n\w*\d+:"`

## 8.83 Tootsville::+Builder-Toot-Hat-Template+

### 8.83.1 Variable

+Builder-Toot-Hat-Template+ names an undocumented variable with the value 2,494 (#x9BE)

## 8.84 Tootsville::+Color24-Names+

### 8.84.1 Variable

+Color24-Names+ names an undocumented variable with the value of type CONS



## 8.85 Tootsville::+Color24-Values+

### 8.85.1 Variable

+Color24-Values+ names an undocumented variable with the value of type CONS

## 8.86 Tootsville::+Credits+

### 8.86.1 Variable

+Credits+ names a variable:

The Tootsville credits

Its value is "Tootsville V by Bruce-Robert Pocock at the Corporation for Inter-World Tourism

Special thanks to Ali Dolan, Mariaelisa Greenwood, Richard Harnden,  
Levi Mc Call, and Zephyr Salz.

In memory of the contributions of Maureen Kenny (RIP).

Tootsville IV by Brandon Booker, Gene Cronk, Robert Dawson, Eric  
Feiling, Tim Hays, Sean King, Mark Mc Corkle, Cassandra Nichol,  
Bruce-Robert Pocock, and Ed Winkelman at Res Interactive, LLC."

## 8.87 Tootsville::+Doc-Packages+

### 8.87.1 Variable

+Doc-Packages+ names a variable:

The packages whose symbols are to be included in the manual.

Its value is of type CONS

## 8.88 Tootsville::+Facing-Angles+

### 8.88.1 Variable

+Facing-Angles+ names a variable:

The eight cardinal directions, mapped to angles in radians.

See Section 8.750 [TOOTSVILLE INTERPRET-FACING], page 1035.

Would be a constant, except for issues with making hash-table constants.

Its value is the hash table:

N	0
NE	0.7853981633974483d0
E	1.5707963267948966d0
SE	2.356194490192345d0
S	3.141592653589793d0
SW	3.9269908169872414d0
W	4.71238898038469d0
NW	5.497787143782138d0

## 8.89 Tootsville::+Gravatar-Base-Uri+

### 8.89.1 Variable

+Gravatar-Base-Uri+ names a variable:

Why would we ever `_not_` use SSL?

Its value is of type `PURI:URI`

## 8.90 Tootsville::+Habitat-Colors+

### 8.90.1 Variable

+Habitat-Colors+ names a variable:

The color triplets which represent each type of habitat in the PNG habitat map.

Its value is of type CONS

## 8.91 Tootsville::+Initial-T-Shirt-Colors+

### 8.91.1 Variable

+Initial-T-Shirt-Colors+ names an undocumented variable with the value of type CONS

## 8.92 Tootsville::+Moon-Day+

### 8.92.1 Variable

+Moon-Day+ names an undocumented variable with the value 63,720 (#xF8E8)



## 8.93 Tootsville::+Moon-Year+

### 8.93.1 Variable

+Moon-Year+ names an undocumented variable with the value 3,600,000 (#x36EE80)

## 8.94 Tootsville::+Other-Moon-Day+

### 8.94.1 Variable

+Other-Moon-Day+ names an undocumented variable with the value 125,280 (#x1E960)

## 8.95 Tootsville::+Other-Moon-Year+

### 8.95.1 Variable

+Other-Moon-Year+ names an undocumented variable with the value 583,243 (#x8E64B)

## 8.96 Tootsville::+Pink-Moon-Day+

### 8.96.1 Variable

+Pink-Moon-Day+ names an undocumented variable with the value 483,840 (#x76200)

## 8.97 Tootsville::+Pink-Moon-Year+

### 8.97.1 Variable

+Pink-Moon-Year+ names an undocumented variable with the value 452,398,723 (#x1AF70E83)

## 8.98 Tootsville::+Pre-Login-Max-Commands+

### 8.98.1 Variable

+Pre-Login-Max-Commands+ names a variable:

How many commands may a client issue before logging in?

This includes the authentication packet. Clients must sign in without issuing an exorbitant number of commands or they will be disconnected.

Its value is 10 (#xA)

## 8.99 Tootsville::+Pre-Login-Max-Time+

### 8.99.1 Variable

+Pre-Login-Max-Time+ names a variable:

How many seconds does a client have to authenticate?

Clients which fail to authenticate within the time limit will be disconnected.

Its value is 5 (#x5)

## 8.100 Tootsville::+Supported-Languages+

### 8.100.1 Variable

+Supported-Languages+ names an undocumented variable with the value of type CONS



## 8.101 Tootsville::+Toot-Base-Color-Names+

### 8.101.1 Variable

+Toot-Base-Color-Names+ names a variable:

Named colors allowed as Toot base colors

Its value is of type CONS

## 8.102 Tootsville::+Toot-Basic-Pattern-Names+

### 8.102.1 Variable

+Toot-Basic-Pattern-Names+ names a variable:

Basic patterns available to any Toot

Its value is of type CONS

## 8.103 Tootsville::+Toot-Extended-Pattern-Names+

### 8.103.1 Variable

+Toot-Extended-Pattern-Names+ names a variable:

Extended patterns that require special effort to obtain

Its value is of type CONS

## 8.104 Tootsville::+Toot-Pad-Color-Names+

### 8.104.1 Variable

+Toot-Pad-Color-Names+ names a variable:

Named colors allowed as Toot pad colors

Its value is of type CONS

## 8.105 Tootsville::+Toot-Pattern-Color-Names+

### 8.105.1 Variable

+Toot-Pattern-Color-Names+ names a variable:

Named colors allowed as Toot pattern colors

Its value is of type CONS

## 8.106 Tootsville::+Unix-Time-In-Universal+

### 8.106.1 Variable

+Unix-Time-In-Universal+ names a variable:

The number of seconds from Universal Time Epoch to Unix Epoch.

Its value is 2,208,988,800 (#x83AA7E80)

## 8.107 Tootsville::+Unix-Zero-In-Universal-Time+

### 8.107.1 Variable

+Unix-Zero-In-Universal-Time+ names a variable:

The Unix zero timestamp occurs at Universal Time 2,208,988,800seconds.

Its value is 2,208,988,800 (#x83AA7E80)

## 8.108 Tootsville::+Ws-Idle-Seconds+

### 8.108.1 Variable

+Ws-Idle-Seconds+ names a variable:

How long before we treat a connection as “idle” and start sending Are You There?

Its value is 300 (#x12C)



## 8.109 Tootsville::2-Days-Ago

### 8.109.1 Function

2-Days-Ago names a function, with lambda list NIL:

Get a timestamp for the second day in the past (the day before yesterday).

### 8.109.2 File

Defined in file `src/types/date+time.lisp`.

## 8.110 Tootsville::3-Days-Ago

### 8.110.1 Function

3-Days-Ago names a function, with lambda list NIL:

Get a timestamp for the third day in the past (the day before the day before yesterday).

### 8.110.2 File

Defined in file `src/types/date+time.lisp`.

## **8.111 Tootsville::@-Message**

### **8.111.1 Function**

Message names an undocumented function, with lambda list (STRING).

### **8.111.2 File**

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.112 Tootsville::Accept-Type-Equal

### 8.112.1 Function

Accept-Type-Equal names an undocumented function, with lambda list (A B &KEY (ALLOW-WILDCARD-P T)).

### 8.112.2 File

Defined in file src/acceptor.lisp.

## 8.113 Tootsville::Acceptor-Status-Message

### 8.113.1 Function

Acceptor-Status-Message names an undocumented function, with lambda list (ACCEPTOR HTTP-STATUS-CODE &REST PROPERTIES &KEY &ALLOW-OTHER-KEYS).

## **8.114 Tootsville::Accepts-Content-Type-P**

### **8.114.1 Function**

Accepts-Content-Type-P names a function, with lambda list (CONTENT-TYPE):

Does the current Hunchentoot request Accept: CONTENT-TYPE?

### **8.114.2 File**

Defined in file src/web.lisp.

## **8.115 Tootsville::Active-Player**

### **8.115.1 Function**

Active-Player names an undocumented function, with lambda list NIL.

### **8.115.2 File**

Defined in file src/websockets.lisp.

## **8.116 Tootsville::Add-Charset**

### **8.116.1 Function**

Add-Charset names a function, with lambda list (CONTENT-TYPE):

Adds the ;charset=UTF-8 type to the end of text and JS/JSON CONTENT-TYPEs

### **8.116.2 File**

Defined in file src/web.lisp.



## 8.117 Tootsville::Add-Contact

### 8.117.1 Function

Add-Contact names an undocumented function, with lambda list (OWNER CONTACT).

### 8.117.2 File

Defined in file src/contacts.lisp.

## 8.118 Tootsville::Add-Or-Replace-Endpoint

### 8.118.1 Function

Add-Or-Replace-Endpoint names an undocumented function, with lambda list (FUNCTION METHOD URI &OPTIONAL CONTENT-TYPE (HOW-SLOW-IS-SLOW)).

## 8.119 Tootsville::Admin-Message

### 8.119.1 Function

Admin-Message names a function, with lambda list (TITLE MESSAGE &KEY (LABEL TITLE)):

Send a broadcast admin MESSAGE with TITLE and LABEL.

Also logs the contents to the console.

### 8.119.2 File

Defined in file src/websockets.lisp.

## 8.120 Tootsville::After-Slash

### 8.120.1 Function

After-Slash names a function, with lambda list (S):

Splits a string S at a slash. Useful for getting the end of a content-type.

Downcases the string. Returns entire string when there's no slash.

### 8.120.2 File

Defined in file src/web.lisp.

## **8.121 Tootsville::All-Connected**

### **8.121.1 Function**

All-Connected names a function, with lambda list NIL:

All clients connected via websockets.

Returns websocket client objects.

### **8.121.2 File**

Defined in file src/websockets.lisp.

## 8.122 Tootsville::All-Credits

### 8.122.1 Function

All-Credits names a function, with lambda list NIL:

Obtain the credits for every system upon which Tootsville is dependant.

Obtains the information from Section 8.361 [TOOTSVILLE DESCRIBE-SYSTEM], page 617, while descending the tree of dependancies from Tootsville through ASDF.

### 8.122.2 File

Defined in file src/main.lisp.

## 8.123 Tootsville::All-Links-To-Same-Person-P

### 8.123.1 Function

All-Links-To-Same-Person-P names an undocumented function, with lambda list (LINKS).

### 8.123.2 File

Defined in file src/users.lisp.

## **8.124 Tootsville::All-Symbols-Alphabetically**

### **8.124.1 Function**

All-Symbols-Alphabetically names a function, with lambda list NIL:

Finds all symbols from Section 8.625 [TOOTSVILLE GATHER-ALL-SYMBOLS], page 883, alphabetically

### **8.124.2 File**

Defined in file src/write-docs-2.lisp.



## 8.125 Tootsville::Allowed-Base-Colors-Under-Pattern

### 8.125.1 Function

Allowed-Base-Colors-Under-Pattern names an undocumented function, with lambda list (&REST ARGUMENTS).

### 8.125.2 File

Defined in file `quicklisp/dists/quicklisp/software/fare-memoization-20180430-git/memoization.lisp`.

## 8.126 Tootsville::Allowed-Pattern-Colors-On-Base

### 8.126.1 Function

Allowed-Pattern-Colors-On-Base names an undocumented function, with lambda list (&REST ARGUMENTS).

### 8.126.2 File

Defined in file `quicklisp/dists/quicklisp/software/fare-memoization-20180430-git/memoization.lisp`.

## **8.127 Tootsville::Altitude**

### **8.127.1 Function**

Altitude names a function, with lambda list (THING):

The altitude of THING

### **8.127.2 File**

Defined in file src/characters/robots.lisp.

## **8.128 Tootsville::Answered-Child-Requests-By-Toot**

### **8.128.1 Function**

Answered-Child-Requests-By-Toot names a function, with lambda list (TOOT):

Recent requests by TOOT to play which have been answered and not expired yet.

### **8.128.2 File**

Defined in file src/users.lisp.

## 8.129 Tootsville::Apply-Config

### 8.129.1 Function

Apply-Config names a function, with lambda list NIL:

Whenever the configuration is loaded, these methods are called to allow “external” packages (which may not use this configuration mechanism) to apply settings.

### 8.129.2 File

Defined in file src/config.lisp.

## 8.130 Tootsville::Apply-Extension-To-Template

### 8.130.1 Function

Apply-Extension-To-Template names a function, with lambda list (TEMPLATE EXTENSION):

Create a clone of TEMPLATE with EXTENSION.

### 8.130.2 File

Defined in file src/web.lisp.

## **8.131 Tootsville::Arrange-Columns+Values-For-Find**

### **8.131.1 Function**

Arrange-Columns+Values-For-Find names an undocumented function, with lambda list (COLUMNS+VALUES COLUMN-DEFINITIONS).

### **8.131.2 File**

Defined in file src/db/db-central.lisp.

## 8.132 Tootsville::Assert-My-Character

### 8.132.1 Function

Assert-My-Character names a function, with lambda list (TOOT-NAME &OPTIONAL (USER \*USER\*)):

Signal a security error if TOOT-NAME is not owned by USER

### 8.132.2 File

Defined in file src/users.lisp.



## 8.133 Tootsville::Associate-Credentials

### 8.133.1 Function

Associate-Credentials names an undocumented function, with lambda list (PERSON CREDENTIALS).

### 8.133.2 File

Defined in file src/users.lisp.

## 8.134 Tootsville::Atom-Or-Comma-List

### 8.134.1 Function

Atom-Or-Comma-List names a function, with lambda list (VALUE):

Return VALUE, possibly by turning it into a comma-delimited string.

An ATOM VALUE is returned intact.

A one-member sequence is returned as the first element of the sequence.

Anything else should be a list that will be turned into a comma-delimited string.

Used in generating HTTP headers.

### 8.134.2 File

Defined in file src/web.lisp.

## 8.135 Tootsville::Avatar

### 8.135.1 Class

Avatar names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.135.2 Slots

Class Avatar has 5 direct slot definitions:

Id

Moniker

Avatar-Scale-X

Avatar-Scale-Y

Avatar-Scale-Z

## **8.136 Tootsville::Avatar-Avatar-Scale-X**

### **8.136.1 Function**

Avatar-Avatar-Scale-X names an undocumented function, with lambda list (INSTANCE).

### **8.136.2 File**

Defined in file src/db/friendly.lisp.

### **8.136.3 SetF Function**

(SETF Avatar-Avatar-Scale-X) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.136.4 File**

Defined in file src/db/friendly.lisp.

## 8.137 Tootsville::Avatar-Avatar-Scale-Y

### 8.137.1 Function

Avatar-Avatar-Scale-Y names an undocumented function, with lambda list (INSTANCE).

### 8.137.2 File

Defined in file src/db/friendly.lisp.

### 8.137.3 SetF Function

(SETF Avatar-Avatar-Scale-Y) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.137.4 File

Defined in file src/db/friendly.lisp.

## **8.138 Tootsville::Avatar-Avatar-Scale-Z**

### **8.138.1 Function**

Avatar-Avatar-Scale-Z names an undocumented function, with lambda list (INSTANCE).

### **8.138.2 File**

Defined in file src/db/friendly.lisp.

### **8.138.3 SetF Function**

(SETF Avatar-Avatar-Scale-Z) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.138.4 File**

Defined in file src/db/friendly.lisp.

## 8.139 Tootsville::Avatar-Has-Slot-P

### 8.139.1 Function

Avatar-Has-Slot-P names an undocumented function, with lambda list (AVATAR SLOT).

### 8.139.2 File

Defined in file src/items.lisp.

## **8.140 Tootsville::Avatar-Id**

### **8.140.1 Function**

Avatar-Id names an undocumented function, with lambda list (INSTANCE).

### **8.140.2 File**

Defined in file src/db/friendly.lisp.

### **8.140.3 SetF Function**

(SETF Avatar-Id) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.140.4 File**

Defined in file src/db/friendly.lisp.



## 8.141 Tootsville::Avatar-Moniker

### 8.141.1 Function

Avatar-Moniker names an undocumented function, with lambda list (INSTANCE).

### 8.141.2 File

Defined in file src/db/friendly.lisp.

### 8.141.3 SetF Function

(SETF Avatar-Moniker) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.141.4 File

Defined in file src/db/friendly.lisp.

## **8.142 Tootsville::Avatar-P**

### **8.142.1 Function**

Avatar-P names an undocumented function, with lambda list (OBJECT).

### **8.142.2 File**

Defined in file src/db/friendly.lisp.

## 8.143 Tootsville::Avatar-Slot

### 8.143.1 Class

Avatar-Slot names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.143.2 Slots

Class Avatar-Slot has 4 direct slot definitions:

Id

Avatar

Slot

Valence

## 8.144 Tootsville::Avatar-Slot-Avatar

### 8.144.1 Function

Avatar-Slot-Avatar names an undocumented function, with lambda list (INSTANCE).

### 8.144.2 File

Defined in file src/db/friendly.lisp.

### 8.144.3 SetF Function

(SETF Avatar-Slot-Avatar) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.144.4 File

Defined in file src/db/friendly.lisp.

## 8.145 Tootsville::Avatar-Slot-Id

### 8.145.1 Function

Avatar-Slot-Id names an undocumented function, with lambda list (INSTANCE).

### 8.145.2 File

Defined in file src/db/friendly.lisp.

### 8.145.3 SetF Function

(SETF Avatar-Slot-Id) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.145.4 File

Defined in file src/db/friendly.lisp.

## **8.146 Tootsville::Avatar-Slot-P**

### **8.146.1 Function**

Avatar-Slot-P names an undocumented function, with lambda list (OBJECT).

### **8.146.2 File**

Defined in file src/db/friendly.lisp.

## 8.147 Tootsville::Avatar-Slot-Slot

### 8.147.1 Function

Avatar-Slot-Slot names an undocumented function, with lambda list (INSTANCE).

### 8.147.2 File

Defined in file src/db/friendly.lisp.

### 8.147.3 SetF Function

(SETF Avatar-Slot-Slot) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.147.4 File

Defined in file src/db/friendly.lisp.

## **8.148 Tootsville::Avatar-Slot-Valence**

### **8.148.1 Function**

Avatar-Slot-Valence names an undocumented function, with lambda list (INSTANCE).

### **8.148.2 File**

Defined in file src/db/friendly.lisp.

### **8.148.3 SetF Function**

(SETF Avatar-Slot-Valence) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.148.4 File**

Defined in file src/db/friendly.lisp.



## **8.149 Tootsville::Average**

### **8.149.1 Function**

Average names an undocumented function, with lambda list (LIST).

### **8.149.2 File**

Defined in file `src/infinity/infinity.lisp`.

## 8.150 Tootsville::Ayt-Idle-Users

### 8.150.1 Function

Ayt-Idle-Users names a function, with lambda list NIL:

Send Are You There to idle (websocket) users.

Idle is defined as idle for Section 8.108 [TOOTSVILLE +WS-IDLE-SECONDS+], page 362, seconds.

### 8.150.2 File

Defined in file src/websockets.lisp.

## 8.151 Tootsville::Background-Gc

### 8.151.1 Function

Background-Gc names a function, with lambda list NIL:

Start a garbage collection in a different thread.

This starts an asynchronous run of the garbage collector, but of course, based on implementation characteristics, this could affect all threads in this image.

Presently only works in SBCL.

### 8.151.2 File

Defined in file src/main.lisp.

## 8.152 Tootsville::Bad-Request

### 8.152.1 Class

Bad-Request names a class, with one superclass: Section 8.667 [TOOTSVILLE HTTP-CLIENT-ERROR], page 925.

A value submitted was the incorrect type, or out of range.

### 8.152.2 Slots

Class Bad-Request has 2 direct slot definitions:

Http-Status-Code

Thing

## 8.153 Tootsville::Bad-Request-Thing

### 8.153.1 Function

Bad-Request-Thing names an undocumented function, with lambda list (CONDITION).

### 8.153.2 SetF Function

(SETF Bad-Request-Thing) names an undocumented function, with lambda list (NEW-VALUE CONDITION).

## **8.154 Tootsville::Banhammer-Ip-Address**

### **8.154.1 Function**

Banhammer-Ip-Address names an undocumented function, with lambda list (ADDRESS).

### **8.154.2 File**

Defined in file src/infinity/legacy-ops.lisp.

## **8.155 Tootsville::Banner**

### **8.155.1 Function**

Banner names a function, with lambda list NIL:

Print greeting banners to the various output streams.

### **8.155.2 File**

Defined in file src/logging.lisp.

## 8.156 Tootsville::Banner/ Error-Output

### 8.156.1 Function

Banner/ Error-Output names a function, with lambda list NIL:

Print a greeting banner to \*ERROR-OUTPUT\* (see the Common Lisp HyperSpec)

### 8.156.2 File

Defined in file src/logging.lisp.



## **8.157 Tootsville::Banner/ Log**

### **8.157.1 Function**

Banner/ Log names a function, with lambda list NIL:

Print a greeting banner to the verbose log.

### **8.157.2 File**

Defined in file src/logging.lisp.

## 8.158 Tootsville::Banner/ Query-Io

### 8.158.1 Function

Banner/ Query-Io names a function, with lambda list NIL:

Print a greeting banner to \*QUERY-IO\* (see the Common Lisp HyperSpec)

### 8.158.2 File

Defined in file src/logging.lisp.

## 8.159 Tootsville::Banner/ Standard-Output

### 8.159.1 Function

Banner/ Standard-Output names a function, with lambda list NIL:

Print a greeting banner to \*STANDARD-OUTPUT\* (see the Common Lisp HyperSpec)

### 8.159.2 File

Defined in file src/logging.lisp.

## 8.160 Tootsville::Banner/ Trace-Output

### 8.160.1 Function

Banner/ Trace-Output names a function, with lambda list NIL:

Print a greeting banner to \*TRACE-OUTPUT\* (see the Common Lisp HyperSpec)

### 8.160.2 File

Defined in file src/logging.lisp.

## **8.161 Tootsville::Base64-From-Uri-Form**

### **8.161.1 Function**

Base64-From-Uri-Form names an undocumented function, with lambda list (TOKEN).

### **8.161.2 File**

Defined in file `src/auth/auth-firebase.lisp`.

## 8.162 Tootsville::Base64-To-Uuid

### 8.162.1 Function

Base64-To-Uuid names a function, with lambda list (VALUE):

Convert a BASE64 value into a UUID.

### 8.162.2 File

Defined in file src/db/db-central.lisp.

## **8.163 Tootsville::Basic-8-Personality**

### **8.163.1 Class**

Basic-8-Personality names a class, with one superclass: Section 8.1269 [TOOTSVILLE TOOT-PERSONALITY], page 1556.

### **8.163.2 Slots**

Class Basic-8-Personality has no direct slots defined.

## 8.164 Tootsville::Before-Save-Normalize

### 8.164.1 Function

Before-Save-Normalize names an undocumented function, with lambda list (OBJECT).



## **8.165 Tootsville::Bool-Sort**

### **8.165.1 Function**

Bool-Sort names a function, with lambda list (A B):

Sort Boolean values

### **8.165.2 File**

Defined in file src/utils.lisp.

## 8.166 Tootsville::Broadcast

### 8.166.1 Function

Broadcast names a function, with lambda list (MESSAGE &KEY NEAR EXCEPT):

Broadcast MESSAGE to all  $\infty$  Mode listeners connected who are near NEAR.

NEAR is a Toot character who is the epicenter of the message, which is currently ignored.

EXCEPT is a user or Toot who does not need to receive the broadcast message (usually the originator)

### 8.166.2 File

Defined in file src/messaging.lisp.

## 8.167 Tootsville::Build-Simple-Column-Query

### 8.167.1 Function

Build-Simple-Column-Query names an undocumented function, with lambda list (TABLE COLUMN COLUMNS).

### 8.167.2 File

Defined in file src/db/maria.lisp.

## 8.168 Tootsville::Build-Simple-Query

### 8.168.1 Function

Build-Simple-Query names an undocumented function, with lambda list (TABLE COLUMNS).

### 8.168.2 File

Defined in file src/db/maria.lisp.

## 8.169 Tootsville::Builder-Toot-P

### 8.169.1 Function

Builder-Toot-P names an undocumented function, with lambda list (&OPTIONAL (TOOT \*TOOT\*)).

### 8.169.2 File

Defined in file src/users.lisp.

## 8.170 Tootsville::Burgeon-Quiesced-State

### 8.170.1 Function

Burgeon-Quiesced-State names a function, with lambda list (TOOT):

Restore quiescent state for TOOT as they return to the game.

### 8.170.2 File

Defined in file `src/infinity/new-commands-20.lisp`.

## 8.171 Tootsville::Byte-Vector-To-Integer

### 8.171.1 Function

Byte-Vector-To-Integer names a function, with lambda list (VECTOR):

Convert VECTOR of (UNSIGNED-BYTE 8) into an integer.

The VECTOR should be in big-endian (aka “network”) byte order.

### 8.171.2 File

Defined in file src/types/binary.lisp.

## **8.172 Tootsville::Bytes-Json**

### **8.172.1 Function**

Bytes-Json names an undocumented function, with lambda list (JSON-BYTES).

### **8.172.2 File**

Defined in file `src/auth/auth-firebase.lisp`.



## 8.173 Tootsville::Call-Infinity-From-Rest

### 8.173.1 Function

Call-Infinity-From-Rest names a function, with lambda list (METHOD):

Call an Infinity-mode command METHOD from a REST call.

Used to create the REST endpoints mapping to METHOD.

### 8.173.2 File

Defined in file src/infinity/infinity.lisp.

## 8.174 Tootsville::Call-Infinity-From-Stream

### 8.174.1 Function

Call-Infinity-From-Stream names a function, with lambda list (JSON):

Call an Infinity-mode command from a stream of JSON packets.

Used by the WebSockets and direct TCP stream handlers.

### 8.174.2 File

Defined in file src/infinity/infinity.lisp.

## **8.175 Tootsville::Cassandra-Add-To-Blacklist**

### **8.175.1 Function**

Cassandra-Add-To-Blacklist names a function, with lambda list (REGEX):

Add REGEX to the blacklist

### **8.175.2 File**

Defined in file src/cassandra.lisp.

## **8.176 Tootsville::Cassandra-Add-To-Redlist**

### **8.176.1 Function**

Cassandra-Add-To-Redlist names a function, with lambda list (REGEX):

Add REGEX to the redlist

### **8.176.2 File**

Defined in file src/cassandra.lisp.

## **8.177 Tootsville::Cassandra-Boot**

### **8.177.1 Function**

Cassandra-Boot names an undocumented function, with lambda list NIL.

### **8.177.2 File**

Defined in file `src/cassandra.lisp`.

## 8.178 Tootsville::Cassandra-Filter

### 8.178.1 Function

Cassandra-Filter names a function, with lambda list (TEXT &OPTIONAL CHILDREN-PRESENT-P):

Filter TEXT for obscenities on the redlist; and, if CHILDREN-PRESENT-P, the blacklist too.

Returns a generalized true value if TEXT should be allowed.

Returns NIL if TEXT should be forbidden.

### 8.178.2 File

Defined in file src/cassandra.lisp.

## 8.179 Tootsville::Cassandra-Obnoxious-Filter

### 8.179.1 Function

Cassandra-Obnoxious-Filter names a function, with lambda list (TEXT VOL):

Filter TEXT for obnoxious content. Starting volume is VOL.

Returns multiple values: the altered versions of TEXT and VOL.

If TEXT is in ‘ALL CAPS LOCK COMPLETELY’, it will be downcased, but VOL will be increased one level (if possible). If TEXT contains certain common repeated or mistyped punctuation, they will be converted.

Note that this is *not* a profanity filter. See Section 8.178 [TOOTSVILLE CASSANDRA-FILTER], page 432, for that feature.

### 8.179.2 File

Defined in file src/cassandra.lisp.

## **8.180 Tootsville::Cassandra-Remove-From-Blacklist**

### **8.180.1 Function**

Cassandra-Remove-From-Blacklist names a function, with lambda list (REGEX):

Remove REGEX from the blacklist.

### **8.180.2 File**

Defined in file src/cassandra.lisp.



## 8.181 Tootsville::Cassandra-Remove-From-Redlist

### 8.181.1 Function

Cassandra-Remove-From-Redlist names a function, with lambda list (REGEX):

Remove REGEX from the redlist.

### 8.181.2 File

Defined in file src/cassandra.lisp.

## **8.182 Tootsville::Chaos-Personality**

### **8.182.1 Class**

Chaos-Personality names a class, with one superclass: Section 8.1074 [TOOTSVILLE ROBOT-CHAOS], page 1359.

This class defines a character named Chaos

### **8.182.2 Slots**

Class Chaos-Personality has no direct slots defined.

## 8.183 Tootsville::Character-Music

### 8.183.1 Class

Character-Music names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.183.2 Slots

Class Character-Music has 2 direct slot definitions:

Music

Toot

## **8.184 Tootsville::Character-Music-Music**

### **8.184.1 Function**

Character-Music-Music names an undocumented function, with lambda list (INSTANCE).

### **8.184.2 File**

Defined in file src/db/friendly.lisp.

### **8.184.3 SetF Function**

(SETF Character-Music-Music) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.184.4 File**

Defined in file src/db/friendly.lisp.

## **8.185 Tootsville::Character-Music-P**

### **8.185.1 Function**

Character-Music-P names an undocumented function, with lambda list (OBJECT).

### **8.185.2 File**

Defined in file src/db/friendly.lisp.

## **8.186 Tootsville::Character-Music-Toot**

### **8.186.1 Function**

Character-Music-Toot names an undocumented function, with lambda list (INSTANCE).

### **8.186.2 File**

Defined in file src/db/friendly.lisp.

### **8.186.3 SetF Function**

(SETF Character-Music-Toot) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.186.4 File**

Defined in file src/db/friendly.lisp.

## **8.187 Tootsville::Chdir**

### **8.187.1 Function**

Chdir names an undocumented function, with lambda list (NEW-PATH).

### **8.187.2 File**

Defined in file src/utils.lisp.

## 8.188 Tootsville::Check-Alexa

### 8.188.1 Function

Check-Alexa names a function, with lambda list (BODY-JSON):

Performs the mandatory checks for queries from Alexa.

Documented by Amazon at: <https://developer.amazon.com/docs/custom-skills/host-a-custom-skill-as-a-web-service.html>

Service

To handle requests sent by Alexa, your web service must meet the following requirements:

1. The service must be Internet-accessible.
2. The service must adhere to the Alexa Skills Kit interface.
3. The service must support HTTP over SSL/TLS, leveraging an Amazon-trusted certificate.
  - For testing, Amazon accepts different methods for providing a certificate. For details, see About the SSL Options.
  - For publishing to end users, Amazon only trusts certificates that have been signed by an Amazon-approved certificate authority.
4. The service must accept requests on port 443.
5. The service must present a certificate with a subject alternate name that matches the domain name of the endpoint.
6. The service must validate that incoming requests are coming from Alexa.

Note: if you are using Apache HTTP Server to host your web service, use version 2.4.10 or later. Earlier versions of Apache HTTP Server send an "unrecognized name" warning if the server is not configured with a `ServerName` or `ServerAlias` in the configuration files. This prevents the Alexa service from sending the customer's request to your server. To address this, either upgrade to 2.4.10 or later, or add `ServerName` / `ServerAlias` to your server's configuration file.

### 8.188.2 File

Defined in file `src/endpoints/gossip/alexa/alexa.lisp`.



## 8.189 Tootsville::Check-Alexa-Signature

### 8.189.1 Function

Check-Alexa-Signature names a function, with lambda list NIL:

Check the signature of an Alexa request.

Excerpt from Amazon requirements at <https://developer.amazon.com/docs/custom-skills/host-a-custom-skill-as-a-web-service.html>:

Checking the Signature of the Request

Requests sent by Alexa provide the information you need to verify the signature in the HTTP headers:

- SignatureCertChainUrl
- Signature

To validate the signature:

1. Verify the URL specified by the SignatureCertChainUrl header value on the request to ensure that it matches the format used by Amazon. See Verifying the Signature Certificate URL.
2. Download the PEM-encoded X.509 certificate chain that Alexa used to sign the message as specified by the SignatureCertChainUrl header value on the request.
3. This chain is provided at runtime so that the certificate may be updated periodically, so your web service should be resilient to different URLs with different content.
4. This certificate chain is composed of, in order, (1) the Amazon signing certificate and (2) one or more additional certificates that create a chain of trust to a root certificate authority (CA) certificate. To confirm the validity of the signing certificate, perform the following checks:
  - The signing certificate has not expired (examine both the Not Before and Not After dates)
  - The domain echo-api.amazon.com is present in the Subject Alternative Names (SANs) section of the signing certificate
  - All certificates in the chain combine to create a chain of trust to a trusted root CA certificate
5. Once you have determined that the signing certificate is valid, extract the public key from it.
6. Base64-decode the Signature header value on the request to obtain the encrypted signature.
7. Use the public key extracted from the signing certificate to decrypt the encrypted signature to produce the asserted hash value.
8. Generate a SHA-1 hash value from the full HTTPS request body to produce the derived hash value
9. Compare the asserted hash value and derived hash values to ensure that they match.

### 8.189.2 File

Defined in file `src/endpoints/gossip/alexa/alexa.lisp`.

## 8.190 Tootsville::Check-Alexa-Signature-Cert-Chain-Url

### 8.190.1 Function

Check-Alexa-Signature-Cert-Chain-Url names a function, with lambda list (URL):

Perform the mandatory checks on an Alexa request's certificate chain URL.

Excerpt from Amazon requirements at <https://developer.amazon.com/docs/custom-skills/host-a-custom-skill-as-a-web-service.html>:

Verifying the Signature Certificate URL

Before downloading the certificate from the URL specified in the SignatureCertChainUrl header, you should ensure that the URL represents a URL Amazon would use for the certificate. This protects against requests that attempt to make your web service download malicious files and similar attacks.

First, normalize the URL so that you can validate against a correctly formatted URL. For example, normalize

`https://s3.amazonaws.com/echo.api/../echo.api/echo-api-cert.pem`

to:

`https://s3.amazonaws.com/echo.api/echo-api-cert.pem`

Next, determine whether the URL meets each of the following criteria:

1. The protocol is equal to `https` (case insensitive).
2. The hostname is equal to `s3.amazonaws.com` (case insensitive).
3. The path starts with `/echo.api/` (case sensitive).
4. If a port is defined in the URL, the port is equal to `443`.

Examples of correctly formatted URLs:

- `https://s3.amazonaws.com/echo.api/echo-api-cert.pem`
- `https://s3.amazonaws.com:443/echo.api/echo-api-cert.pem`
- `https://s3.amazonaws.com/echo.api/../echo.api/echo-api-cert.pem`

Examples of invalid URLs:

- `http://s3.amazonaws.com/echo.api/echo-api-cert.pem` (invalid protocol)
- `https://notamazon.com/echo.api/echo-api-cert.pem` (invalid hostname)
- `https://s3.amazonaws.com/EcHo.aPi/echo-api-cert.pem` (invalid path)
- `https://s3.amazonaws.com/invalid.path/echo-api-cert.pem` (invalid path)
- `https://s3.amazonaws.com:563/echo.api/echo-api-cert.pem` (invalid port)

If the URL does not pass these tests, reject the request and do not proceed with verifying the signature.

### 8.190.2 File

Defined in file `src/endpoints/gossip/alexa/alexa.lisp`.

## 8.191 Tootsville::Check-Alexa-Timestamp-Tolerance

### 8.191.1 Function

Check-Alexa-Timestamp-Tolerance names a function, with lambda list (BODY-JSON):

Ensure that the timestamp of an Alexa-sent query is within the allowed tolerance.

Excerpt from Amazon requirements at <https://developer.amazon.com/docs/custom-skills/host-a-custom-skill-as-a-web-service.html>:

Checking the Timestamp of the Request

Every request sent to your web service by Alexa includes a timestamp. This information is part of the signed portion of the request, so it cannot be changed without also invalidating the request signature. Using this timestamp to verify the freshness of the request before responding protects your service from attackers attempting a "replay" attack in which they acquire a properly signed request and then repeatedly resend it to disrupt your service.

Your service should allow a tolerance of no more than 150 seconds (two and a half minutes). This means that your service should only accept requests in which the request timestamp is within 150 seconds of the current time. Web services that allow a longer tolerance cannot be published to Amazon customers.

...

If you are not using the Java library, you need to do this verification yourself. The timestamp is provided as part of the request object in the JSON body of the request ... The timestamp is provided as an ISO 8601 formatted string (for example, 2015-05-13T12:34:56Z). Your code needs to parse the string into a date object, then verify that it is within the tolerance your web service allows (no more than 150 seconds). Reject requests in which the timestamp falls outside the tolerance with an error code (such as 400 Bad Request).

### 8.191.2 File

Defined in file `src/endpoints/gossip/alexa/alexa.lisp`.

## 8.192 Tootsville::Check-Arg-Type

### 8.192.1 Macro

Check-Arg-Type names a macro, with lambda list (ARG TYPE &OPTIONAL NAME):

Ensure that ARG is of type TYPE, which is called NAME. Signals back to an HTTP client with a 400 error if this assertion is untrue.

This is basically just CHECK-TYPE for arguments passed by the user.

### 8.192.2 File

Defined in file src/web.lisp.

## **8.193 Tootsville::Check-Cert-Chain-Valid**

### **8.193.1 Function**

Check-Cert-Chain-Valid names an undocumented function, with lambda list (X.509-CERT).

### **8.193.2 File**

Defined in file `src/endpoints/gossip/alexa/alexa.lisp`.

## **8.194 Tootsville::Check-Cert-Dates-Valid**

### **8.194.1 Function**

Check-Cert-Dates-Valid names an undocumented function, with lambda list (X.509-CERT).

### **8.194.2 File**

Defined in file `src/endpoints/gossip/alexa/alexa.lisp`.

## 8.195 Tootsville::Check-Firebase-Id-Token

### 8.195.1 Function

Check-Firebase-Id-Token names an undocumented function, with lambda list (TOKEN).

### 8.195.2 File

Defined in file `src/auth/auth-firebase.lisp`.

## 8.196 Tootsville::Check-Pattern-On-Base-Color

### 8.196.1 Function

Check-Pattern-On-Base-Color names a function, with lambda list (PATTERN-COLOR BASE-COLOR &KEY TOOT-NAME PAD-COLOR PATTERN ADDRESS):

Ensure that the PATTERN-COLOR is allowed on the BASE-COLOR.

TOOT-NAME, PAD-COLOR, PATTERN, and ADDRESS are used to provide additional error details.

Provides restarts CHANGE-PATTERN-COLOR and CHANGE-BASE-COLOR to correct any deficiencies.

### 8.196.2 File

Defined in file src/types/color+pattern.lisp.



## 8.197 Tootsville::Check-Toot-Name

### 8.197.1 Function

Check-Toot-Name names a function, with lambda list (NAME):

Check if NAME is allowed as a Toot name; offering restarts to correct it, if not.

This is generally intended for accepting new Toot names, versus validating REST calls, for example.

### 8.197.2 File

Defined in file `src/types/toot-names.lisp`.

## **8.198 Tootsville::Check-X.509-San**

### **8.198.1 Function**

Check-X.509-San names a function, with lambda list (X.509-CERT NAME):

Ensure that NAME is a DNS Alt Name for the subject of the X.509-CERT

### **8.198.2 File**

Defined in file src/endpoints/gossip/alexandria/alexandria.lisp.

## **8.199 Tootsville::Child-Code**

### **8.199.1 Type**

Child-Code names a TYPE:

A potential child code (password).

See Section 8.1324 [TOOTSVILLE VALID-CHILD-CODE-P], page 1611.

## 8.200 Tootsville::Child-Request

### 8.200.1 Class

Child-Request names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.200.2 Slots

Class Child-Request has 7 direct slot definitions:

Uuid

Toot

Placed-At

Allowed-At

Denied-At

Allowed-For

Response

## 8.201 Tootsville::Child-Request-Allowed-At

### 8.201.1 Function

Child-Request-Allowed-At names an undocumented function, with lambda list (INSTANCE).

### 8.201.2 File

Defined in file src/db/friendly.lisp.

### 8.201.3 SetF Function

(SETF Child-Request-Allowed-At) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.201.4 File

Defined in file src/db/friendly.lisp.

## 8.202 Tootsville::Child-Request-Allowed-For

### 8.202.1 Function

Child-Request-Allowed-For names an undocumented function, with lambda list (INSTANCE).

### 8.202.2 File

Defined in file src/db/friendly.lisp.

### 8.202.3 SetF Function

(SETF Child-Request-Allowed-For) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.202.4 File

Defined in file src/db/friendly.lisp.

## 8.203 Tootsville::Child-Request-Allowed-Until

### 8.203.1 Function

Child-Request-Allowed-Until names an undocumented function, with lambda list (REQUEST).

### 8.203.2 File

Defined in file src/users.lisp.

## 8.204 Tootsville::Child-Request-Denied-At

### 8.204.1 Function

Child-Request-Denied-At names an undocumented function, with lambda list (INSTANCE).

### 8.204.2 File

Defined in file src/db/friendly.lisp.

### 8.204.3 SetF Function

(SETF Child-Request-Denied-At) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.204.4 File

Defined in file src/db/friendly.lisp.



## **8.205 Tootsville::Child-Request-P**

### **8.205.1 Function**

Child-Request-P names an undocumented function, with lambda list (OBJECT).

### **8.205.2 File**

Defined in file src/db/friendly.lisp.

## **8.206 Tootsville::Child-Request-Placed-At**

### **8.206.1 Function**

Child-Request-Placed-At names an undocumented function, with lambda list (INSTANCE).

### **8.206.2 File**

Defined in file src/db/friendly.lisp.

### **8.206.3 SetF Function**

(SETF Child-Request-Placed-At) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.206.4 File**

Defined in file src/db/friendly.lisp.

## 8.207 Tootsville::Child-Request-Response

### 8.207.1 Function

Child-Request-Response names an undocumented function, with lambda list (INSTANCE).

### 8.207.2 File

Defined in file src/db/friendly.lisp.

### 8.207.3 SetF Function

(SETF Child-Request-Response) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.207.4 File

Defined in file src/db/friendly.lisp.

## **8.208 Tootsville::Child-Request-Toot**

### **8.208.1 Function**

Child-Request-Toot names an undocumented function, with lambda list (INSTANCE).

### **8.208.2 File**

Defined in file src/db/friendly.lisp.

### **8.208.3 SetF Function**

(SETF Child-Request-Toot) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.208.4 File**

Defined in file src/db/friendly.lisp.

## 8.209 Tootsville::Child-Request-Uuid

### 8.209.1 Function

Child-Request-Uuid names an undocumented function, with lambda list (INSTANCE).

### 8.209.2 File

Defined in file src/db/friendly.lisp.

### 8.209.3 SetF Function

(SETF Child-Request-Uuid) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.209.4 File

Defined in file src/db/friendly.lisp.

## 8.210 Tootsville::Clean-Ice-Credentials

### 8.210.1 Function

Clean-Ice-Credentials names an undocumented function, with lambda list (CREDENTIALS).

### 8.210.2 File

Defined in file src/gossip.lisp.

## 8.211 Tootsville::Clean-Symbols

### 8.211.1 Function

Clean-Symbols names an undocumented function, with lambda list (SRC).

### 8.211.2 File

Defined in file src/write-docs-2.lisp.

## 8.212 Tootsville::Clear-All-Endpoints

### 8.212.1 Function

Clear-All-Endpoints names a function, with lambda list NIL:

Destroy every endpoint mapping.

Leaves the functions intact. Useful for debugging; should probably never run in production.

### 8.212.2 File

Defined in file `src/endpoint.lisp`.



## **8.213 Tootsville::Clouds**

### **8.213.1 Function**

Clouds names a function, with lambda list (X Y Z):

The current cloud cover at X,Y,Z

### **8.213.2 File**

Defined in file src/weather/weather.lisp.

## 8.214 Tootsville::Cluster

### 8.214.1 Function

Cluster names a function, with lambda list NIL:

Get the identity of the current cluster.

Returns one of:

- :test
- :qa
- :prod

### 8.214.2 File

Defined in file src/config.lisp.

## 8.215 Tootsville::Cluster-Name

### 8.215.1 Function

Cluster-Name names a function, with lambda list (&OPTIONAL PREFIX):

Get the name of the active cluster.

Currently one of:

- test.tootsville.org
- qa.tootsville.org
- tootsville.org

The local hostname is used in development (loopback) mode.

### 8.215.2 File

Defined in file src/config.lisp.

## 8.216 Tootsville::Cluster-Net-Name

### 8.216.1 Function

Cluster-Net-Name names an undocumented function, with lambda list (&OPTIONAL PREFIX).

### 8.216.2 File

Defined in file src/config.lisp.

## 8.217 Tootsville::Cluster-Wide-Lock-Busy-Error

### 8.217.1 Class

Cluster-Wide-Lock-Busy-Error names a class, with one superclass: Section 8.220 [TOOTSVILLE CLUSTER-WIDE-LOCK-ERROR], page 474.

### 8.217.2 Slots

Class Cluster-Wide-Lock-Busy-Error has no direct slots defined.

## **8.218 Tootsville::Cluster-Wide-Lock-Busy-Warning**

### **8.218.1 Class**

Cluster-Wide-Lock-Busy-Warning names a class, with two superclasses: Section 8.219 [TOOTSVILLE CLUSTER-WIDE-LOCK-CONDITION], page 473, and COMMON-LISP::WARNING (not in this manual)

### **8.218.2 Slots**

Class Cluster-Wide-Lock-Busy-Warning has no direct slots defined.

## 8.219 Tootsville::Cluster-Wide-Lock-Condition

### 8.219.1 Class

Cluster-Wide-Lock-Condition names a class, with one superclass: COMMON-LISP::SERIOUS-CONDITION (not in this manual).

### 8.219.2 Slots

Class Cluster-Wide-Lock-Condition has no direct slots defined.

## 8.220 Tootsville::Cluster-Wide-Lock-Error

### 8.220.1 Class

Cluster-Wide-Lock-Error names a class, with two superclasses: Section 8.219 [TOOTSVILLE CLUSTER-WIDE-LOCK-CONDITION], page 473, and COMMON-LISP::ERROR (not in this manual)

### 8.220.2 Slots

Class Cluster-Wide-Lock-Error has no direct slots defined.



## 8.221 Tootsville::Cluster-Wide-Lock-Not-Locked

### 8.221.1 Class

Cluster-Wide-Lock-Not-Locked names a class, with two superclasses: Section 8.219 [TOOTSVILLE CLUSTER-WIDE-LOCK-CONDITION], page 473, and COMMON-LISP::WARNING (not in this manual)

### 8.221.2 Slots

Class Cluster-Wide-Lock-Not-Locked has no direct slots defined.

## 8.222 Tootsville::Cluster-Wide-Lock-Not-Ours

### 8.222.1 Class

Cluster-Wide-Lock-Not-Ours names a class, with one superclass: Section 8.220 [TOOTSVILLE CLUSTER-WIDE-LOCK-ERROR], page 474.

### 8.222.2 Slots

Class Cluster-Wide-Lock-Not-Ours has no direct slots defined.

## 8.223 Tootsville::Color24

### 8.223.1 Class

Color24 names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.223.2 Slots

Class Color24 has 3 direct slot definitions:

Red

Green

Blue

## 8.224 Tootsville::Color24-Blue

### 8.224.1 Function

Color24-Blue names an undocumented function, with lambda list (INSTANCE).

### 8.224.2 File

Defined in file src/types/color+pattern.lisp.

### 8.224.3 SetF Function

(SETF Color24-Blue) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.224.4 File

Defined in file src/types/color+pattern.lisp.

## 8.225 Tootsville::Color24-Green

### 8.225.1 Function

Color24-Green names an undocumented function, with lambda list (INSTANCE).

### 8.225.2 File

Defined in file src/types/color+pattern.lisp.

### 8.225.3 SetF Function

(SETF Color24-Green) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.225.4 File

Defined in file src/types/color+pattern.lisp.

## 8.226 Tootsville::Color24-Hsv

### 8.226.1 Function

Color24-Hsv names a function, with lambda list (COLOR):

Extract the Hue, Saturation, Value of a Color24 as a list of reals 0...1

### 8.226.2 File

Defined in file src/types/color+pattern.lisp.

## 8.227 Tootsville::Color24-Hue

### 8.227.1 Function

Color24-Hue names a function, with lambda list (COLOR):

The HSV Hue channel of COLOR.

See Section 8.226 [TOOTSVILLE COLOR24-HSV], page 480.

### 8.227.2 File

Defined in file src/types/color+pattern.lisp.

## 8.228 Tootsville::Color24-Name

### 8.228.1 Function

Color24-Name names a function, with lambda list (COLOR):

Given COLOR, return the name or hex string for it.

If COLOR is a named color in Section 8.85 [TOOTSVILLE +COLOR24-VALUES+], page 339, returns its name. Otherwise, returns the 6-digit hex value, as per HTML or CSS.

### 8.228.2 File

Defined in file src/types/color+pattern.lisp.



## **8.229 Tootsville::Color24-P**

### **8.229.1 Function**

Color24-P names an undocumented function, with lambda list (OBJECT).

### **8.229.2 File**

Defined in file src/types/color+pattern.lisp.

## **8.230 Tootsville::Color24-Red**

### **8.230.1 Function**

Color24-Red names an undocumented function, with lambda list (INSTANCE).

### **8.230.2 File**

Defined in file src/types/color+pattern.lisp.

### **8.230.3 SetF Function**

(SETF Color24-Red) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.230.4 File**

Defined in file src/types/color+pattern.lisp.

## 8.231 Tootsville::Color24-Rgb

### 8.231.1 Function

Color24-Rgb names a function, with lambda list (R G B):

Construct a Color24 from a R G B triplet

### 8.231.2 File

Defined in file src/types/color+pattern.lisp.

## 8.232 Tootsville::Color24-Saturation

### 8.232.1 Function

Color24-Saturation names a function, with lambda list (COLOR):

The HSV Saturation channel of COLOR.

See Section 8.226 [TOOTSVILLE COLOR24-HSV], page 480.

### 8.232.2 File

Defined in file src/types/color+pattern.lisp.

## 8.233 Tootsville::Color24-To-Integer

### 8.233.1 Function

Color24-To-Integer names a function, with lambda list (COLOR):

Return a 24-bit integer representing COLOR.

The upper 8 bits are the red channel; the next 8 bits, green; and the lowest 8 bits, the blue channel.

### 8.233.2 File

Defined in file `src/types/color+pattern.lisp`.

## 8.234 Tootsville::Color24-Value

### 8.234.1 Function

Color24-Value names a function, with lambda list (COLOR):

The HSV Value channel of COLOR.

See Section 8.226 [TOOTSVILLE COLOR24-HSV], page 480.

### 8.234.2 File

Defined in file src/types/color+pattern.lisp.

## 8.235 Tootsville::Color24/ =

### 8.235.1 Function

Color24/ = names a function, with lambda list (A B):

Comparator of two color24s

### 8.235.2 File

Defined in file src/types/color+pattern.lisp.

## 8.236 Tootsville::Color24=

### 8.236.1 Function

Color24= names a function, with lambda list (A B &REST MORE):

Comparator of two or more Section 8.223 [TOOTSVILLE COLOR24], page 477, values.

Values are the same only if all three channels (red, green, and blue) have identical values.

### 8.236.2 File

Defined in file src/types/color+pattern.lisp.



## 8.237 Tootsville::Column-Load-Mapping

### 8.237.1 Function

Column-Load-Mapping names a function, with lambda list (COLUMN):

Map COLUMN from a database record into internal form.

Used in Section 8.340 [TOOTSVILLE DEFRECORD], page 596, qv.

### 8.237.2 File

Defined in file src/db/db-central.lisp.

## 8.238 Tootsville::Column-Load-Value

### 8.238.1 Function

Column-Load-Value names a function, with lambda list (VALUE TYPE):

For a column of TYPE, interpret raw VALUE

### 8.238.2 File

Defined in file src/db/db-central.lisp.

## 8.239 Tootsville::Column-Normalizer

### 8.239.1 Function

Column-Normalizer names an undocumented function, with lambda list (COLUMN).

### 8.239.2 File

Defined in file src/db/db-central.lisp.

## **8.240 Tootsville::Column-Save-Mapping**

### **8.240.1 Function**

Column-Save-Mapping names an undocumented function, with lambda list (COLUMN).

### **8.240.2 File**

Defined in file `src/db/db-central.lisp`.

## 8.241 Tootsville::Column-Save-Value

### 8.241.1 Function

Column-Save-Value names a function, with lambda list (VALUE TYPE):

Convert VALUE into the database's representation of TYPE

### 8.241.2 File

Defined in file src/db/db-central.lisp.

## **8.242 Tootsville::Compute-Fountain-Peanuts-For-Score**

### **8.242.1 Function**

Compute-Fountain-Peanuts-For-Score names an undocumented function, with lambda list (SCORE).

### **8.242.2 File**

Defined in file src/quaestor.lisp.

## 8.243 Tootsville::Compute-Fountain-Random-Fairy-Dust

### 8.243.1 Function

Compute-Fountain-Random-Fairy-Dust names an undocumented function, with lambda list NIL.

### 8.243.2 File

Defined in file src/quaestor.lisp.

## 8.244 Tootsville::Compute-Next-Keys-Update

### 8.244.1 Function

Compute-Next-Keys-Update names an undocumented function, with lambda list (HEADERS-ALIST).

### 8.244.2 File

Defined in file src/auth/auth-firebase.lisp.



## **8.245 Tootsville::Concat**

### **8.245.1 Function**

Concat names an undocumented function, with lambda list (&REST ARGS).

### **8.245.2 File**

Defined in file `src/endpoints/slash-meta-game.lisp`.

## 8.246 Tootsville::Condition-Name

### 8.246.1 Function

Condition-Name names a function, with lambda list (CONDITION):

Returns the capitalized name of the class of CONDITION.

### 8.246.2 File

Defined in file src/errors.lisp.

## 8.247 Tootsville::Condition-Slots

### 8.247.1 Function

Condition-Slots names a function, with lambda list (OBJECT):

Enumerates the name of every slot on OBJECT

### 8.247.2 File

Defined in file src/errors.lisp.

## **8.248 Tootsville::Config**

### **8.248.1 Function**

Config names a function, with lambda list (&REST KEYS):

Obtain the configuration value at the path KEY + SUB-KEYS

### **8.248.2 File**

Defined in file src/config.lisp.

## 8.249 Tootsville::Connect-Cache

### 8.249.1 Function

Connect-Cache names a function, with lambda list NIL:

Connect to MemCached.

Configuration comes from Section 8.317 [TOOTSVILLE DB-CONFIG], page 571, path (:cache (:ip :port :name)).

The pool size will be OLIPHAUNT::PROCESSOR-COUNT (not in this manual), clamped to 3-15.

### 8.249.2 File

Defined in file src/db/memcached.lisp.

## **8.250 Tootsville::Connect-Databases**

### **8.250.1 Function**

Connect-Databases names a function, with lambda list NIL:

Connect all database systems in parallel (each in its own thread)

### **8.250.2 File**

Defined in file src/main.lisp.

## 8.251 Tootsville::Connect-Maria

### 8.251.1 Function

Connect-Maria names a function, with lambda list NIL:

Make a connection to MariaDB.

This ensures that it is reachable, and that there is at least one connection in the pool.

### 8.251.2 File

Defined in file src/db/maria.lisp.

## **8.252 Tootsville::Connect-Time**

### **8.252.1 Function**

Connect-Time names an undocumented function, with lambda list (OBJECT).



## 8.253 Tootsville::Connected-Toot-Names

### 8.253.1 Function

Connected-Toot-Names names a function, with lambda list NIL:

The names of all Toots currently connected — players or NPCs.

### 8.253.2 File

Defined in file src/websockets.lisp.

## **8.254 Tootsville::Connected-Toots**

### **8.254.1 Function**

Connected-Toots names a function, with lambda list NIL:

All Toots currently connected — players or NPCs

### **8.254.2 File**

Defined in file src/websockets.lisp.

## 8.255 Tootsville::Consider-Child-Kick

### 8.255.1 Function

Consider-Child-Kick names a function, with lambda list (TOOT):

Decide whether to kick TOOT offline due to time expiring.

If there is no approved request for TOOT to continue in the game, they'll be kicked offline. If there is, then we'll schedule to recheck this when that time is elapsed.

Calling this with an adult's Toot is funny, but not helpful.

### 8.255.2 File

Defined in file `src/websockets.lisp`.

## 8.256 Tootsville::Constituentp

### 8.256.1 Function

Constituentp names a function, with lambda list (CH):

Is character CH a constituent character of a Lisp name (without quoting)?

Accepts A-Z, 0-9, any character above #xa0, and these punctuation: -/!%.

### 8.256.2 File

Defined in file src/web.lisp.

## 8.257 Tootsville::Contact

### 8.257.1 Class

Contact names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.257.2 Slots

Class Contact has 6 direct slot definitions:

Uuid

Owner

Contact

Starredp

Added

Last-Used

## **8.258 Tootsville::Contact-Added**

### **8.258.1 Function**

Contact-Added names an undocumented function, with lambda list (INSTANCE).

### **8.258.2 File**

Defined in file src/db/friendly.lisp.

### **8.258.3 SetF Function**

(SETF Contact-Added) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.258.4 File**

Defined in file src/db/friendly.lisp.

## 8.259 Tootsville::Contact-Contact

### 8.259.1 Function

Contact-Contact names an undocumented function, with lambda list (INSTANCE).

### 8.259.2 File

Defined in file src/db/friendly.lisp.

### 8.259.3 SetF Function

(SETF Contact-Contact) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.259.4 File

Defined in file src/db/friendly.lisp.

## **8.260 Tootsville::Contact-Last-Used**

### **8.260.1 Function**

Contact-Last-Used names an undocumented function, with lambda list (INSTANCE).

### **8.260.2 File**

Defined in file src/db/friendly.lisp.

### **8.260.3 SetF Function**

(SETF Contact-Last-Used) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.260.4 File**

Defined in file src/db/friendly.lisp.



## 8.261 Tootsville::Contact-Owner

### 8.261.1 Function

Contact-Owner names an undocumented function, with lambda list (INSTANCE).

### 8.261.2 File

Defined in file src/db/friendly.lisp.

### 8.261.3 SetF Function

(SETF Contact-Owner) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.261.4 File

Defined in file src/db/friendly.lisp.

## **8.262 Tootsville::Contact-P**

### **8.262.1 Function**

Contact-P names an undocumented function, with lambda list (OBJECT).

### **8.262.2 File**

Defined in file src/db/friendly.lisp.

## 8.263 Tootsville::Contact-Starredp

### 8.263.1 Function

Contact-Starredp names an undocumented function, with lambda list (INSTANCE).

### 8.263.2 File

Defined in file src/db/friendly.lisp.

### 8.263.3 SetF Function

(SETF Contact-Starredp) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.263.4 File

Defined in file src/db/friendly.lisp.

## **8.264 Tootsville::Contact-Uuid**

### **8.264.1 Function**

Contact-Uuid names an undocumented function, with lambda list (INSTANCE).

### **8.264.2 File**

Defined in file src/db/friendly.lisp.

### **8.264.3 SetF Function**

(SETF Contact-Uuid) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.264.4 File**

Defined in file src/db/friendly.lisp.

## 8.265 Tootsville::Contents-To-Bytes

### 8.265.1 Function

Contents-To-Bytes names a function, with lambda list (CONTENTS):

Convert CONTENTS to a sequence of 8-bit bytes.

Assumes strings are UTF-8; vectors are already bytes; and lists are JSON faux data.

### 8.265.2 File

Defined in file src/web.lisp.

## **8.266 Tootsville::Copy-Avatar**

### **8.266.1 Function**

Copy-Avatar names an undocumented function, with lambda list (INSTANCE).

### **8.266.2 File**

Defined in file src/db/friendly.lisp.

## **8.267 Tootsville::Copy-Avatar-Slot**

### **8.267.1 Function**

Copy-Avatar-Slot names an undocumented function, with lambda list (INSTANCE).

### **8.267.2 File**

Defined in file src/db/friendly.lisp.

## **8.268 Tootsville::Copy-Character-Music**

### **8.268.1 Function**

Copy-Character-Music names an undocumented function, with lambda list (INSTANCE).

### **8.268.2 File**

Defined in file src/db/friendly.lisp.



## 8.269 Tootsville::Copy-Child-Request

### 8.269.1 Function

Copy-Child-Request names an undocumented function, with lambda list (INSTANCE).

### 8.269.2 File

Defined in file src/db/friendly.lisp.

## **8.270 Tootsville::Copy-Color24**

### **8.270.1 Function**

Copy-Color24 names an undocumented function, with lambda list (INSTANCE).

### **8.270.2 File**

Defined in file `src/types/color+pattern.lisp`.

## **8.271 Tootsville::Copy-Contact**

### **8.271.1 Function**

Copy-Contact names an undocumented function, with lambda list (INSTANCE).

### **8.271.2 File**

Defined in file src/db/friendly.lisp.

## **8.272 Tootsville::Copy-Credential**

### **8.272.1 Function**

Copy-Credential names an undocumented function, with lambda list (INSTANCE).

### **8.272.2 File**

Defined in file src/db/friendly.lisp.

## 8.273 Tootsville::Copy-Game-Point

### 8.273.1 Function

Copy-Game-Point names an undocumented function, with lambda list (INSTANCE).

### 8.273.2 File

Defined in file src/characters/robots.lisp.

## **8.274 Tootsville::Copy-Gossip-Initiation**

### **8.274.1 Function**

Copy-Gossip-Initiation names an undocumented function, with lambda list (INSTANCE).

### **8.274.2 File**

Defined in file src/gossip.lisp.

## 8.275 Tootsville::Copy-Inventory-Item

### 8.275.1 Function

Copy-Inventory-Item names an undocumented function, with lambda list (INSTANCE).

### 8.275.2 File

Defined in file src/db/friendly.lisp.

## **8.276 Tootsville::Copy-Item**

### **8.276.1 Function**

Copy-Item names an undocumented function, with lambda list (INSTANCE).

### **8.276.2 File**

Defined in file src/db/friendly.lisp.



## 8.277 Tootsville::Copy-Item-Template

### 8.277.1 Function

Copy-Item-Template names an undocumented function, with lambda list (INSTANCE).

### 8.277.2 File

Defined in file src/db/friendly.lisp.

## **8.278 Tootsville::Copy-Locale-Music**

### **8.278.1 Function**

Copy-Locale-Music names an undocumented function, with lambda list (INSTANCE).

### **8.278.2 File**

Defined in file src/db/friendly.lisp.

## **8.279 Tootsville::Copy-Login**

### **8.279.1 Function**

Copy-Login names an undocumented function, with lambda list (INSTANCE).

### **8.279.2 File**

Defined in file src/db/friendly.lisp.

## **8.280 Tootsville::Copy-Lot**

### **8.280.1 Function**

Copy-Lot names an undocumented function, with lambda list (INSTANCE).

### **8.280.2 File**

Defined in file src/db/friendly.lisp.

## 8.281 Tootsville::Copy-Metronome-Task

### 8.281.1 Function

Copy-Metronome-Task names an undocumented function, with lambda list (INSTANCE).

### 8.281.2 File

Defined in file src/metronome.lisp.

## **8.282 Tootsville::Copy-Mist**

### **8.282.1 Function**

Copy-Mist names an undocumented function, with lambda list (INSTANCE).

### **8.282.2 File**

Defined in file src/db/friendly.lisp.

## **8.283 Tootsville::Copy-Music**

### **8.283.1 Function**

Copy-Music names an undocumented function, with lambda list (INSTANCE).

### **8.283.2 File**

Defined in file src/db/friendly.lisp.

## **8.284 Tootsville::Copy-Parent-Child**

### **8.284.1 Function**

Copy-Parent-Child names an undocumented function, with lambda list (INSTANCE).

### **8.284.2 File**

Defined in file src/db/friendly.lisp.



## 8.285 Tootsville::Copy-Pattern

### 8.285.1 Function

Copy-Pattern names an undocumented function, with lambda list (INSTANCE).

### 8.285.2 File

Defined in file src/db/friendly.lisp.

## **8.286 Tootsville::Copy-Person**

### **8.286.1 Function**

Copy-Person names an undocumented function, with lambda list (INSTANCE).

### **8.286.2 File**

Defined in file src/db/friendly.lisp.

## **8.287 Tootsville::Copy-Person-Link**

### **8.287.1 Function**

Copy-Person-Link names an undocumented function, with lambda list (INSTANCE).

### **8.287.2 File**

Defined in file src/db/friendly.lisp.

## **8.288 Tootsville::Copy-Place**

### **8.288.1 Function**

Copy-Place names an undocumented function, with lambda list (INSTANCE).

### **8.288.2 File**

Defined in file src/db/friendly.lisp.

## 8.289 Tootsville::Copy-Quaestor-Event

### 8.289.1 Function

Copy-Quaestor-Event names an undocumented function, with lambda list (INSTANCE).

### 8.289.2 File

Defined in file src/db/friendly.lisp.

## **8.290 Tootsville::Copy-Sms**

### **8.290.1 Function**

Copy-Sms names an undocumented function, with lambda list (INSTANCE).

### **8.290.2 File**

Defined in file src/db/friendly.lisp.

## **8.291 Tootsville::Copy-Store-Item**

### **8.291.1 Function**

Copy-Store-Item names an undocumented function, with lambda list (INSTANCE).

### **8.291.2 File**

Defined in file src/db/friendly.lisp.

## **8.292 Tootsville::Copy-Tcp-Client**

### **8.292.1 Function**

Copy-Tcp-Client names an undocumented function, with lambda list (INSTANCE).

### **8.292.2 File**

Defined in file src/tcp-stream.lisp.



## 8.293 Tootsville::Copy-Terrain-Edge-Horz

### 8.293.1 Function

Copy-Terrain-Edge-Horz names an undocumented function, with lambda list (START-LATITUDE LONGITUDE END-LATITUDE DEST-LATITUDE DEST-LONGITUDE).

### 8.293.2 File

Defined in file src/terrain.lisp.

## 8.294 Tootsville::Copy-Terrain-Edge-Vert

### 8.294.1 Function

Copy-Terrain-Edge-Vert names an undocumented function, with lambda list (LATITUDE START-LONGITUDE END-LONGITUDE DEST-LATITUDE DEST-LONGITUDE).

### 8.294.2 File

Defined in file src/terrain.lisp.

## 8.295 Tootsville::Copy-Terrain-Height

### 8.295.1 Function

Copy-Terrain-Height names an undocumented function, with lambda list (INSTANCE).

### 8.295.2 File

Defined in file src/db/friendly.lisp.

## **8.296 Tootsville::Copy-Toot**

### **8.296.1 Function**

Copy-Toot names an undocumented function, with lambda list (INSTANCE).

### **8.296.2 File**

Defined in file src/db/friendly.lisp.

## 8.297 Tootsville::Copy-Toot-Quiesced

### 8.297.1 Function

Copy-Toot-Quiesced names an undocumented function, with lambda list (INSTANCE).

### 8.297.2 File

Defined in file src/db/friendly.lisp.

## **8.298 Tootsville::Copy-Wear-Slot**

### **8.298.1 Function**

Copy-Wear-Slot names an undocumented function, with lambda list (INSTANCE).

### **8.298.2 File**

Defined in file src/db/friendly.lisp.

## 8.299 Tootsville::Copy-Wind-Vector

### 8.299.1 Function

Copy-Wind-Vector names an undocumented function, with lambda list (INSTANCE).

### 8.299.2 File

Defined in file src/weather/weather.lisp.

## **8.300 Tootsville::Copy-World**

### **8.300.1 Function**

Copy-World names an undocumented function, with lambda list (INSTANCE).

### **8.300.2 File**

Defined in file src/db/friendly.lisp.



## **8.301 Tootsville::Copy-Wtl-Course**

### **8.301.1 Function**

Copy-Wtl-Course names an undocumented function, with lambda list (INSTANCE).

### **8.301.2 File**

Defined in file `src/characters/robots.lisp`.

## 8.302 Tootsville::Create-Item

### 8.302.1 Function

Create-Item names a function, with lambda list (TEMPLATE-ID):

Create an item as an instance of the given TEMPLATE-ID.

### 8.302.2 File

Defined in file src/items.lisp.

## 8.303 Tootsville::Credential

### 8.303.1 Class

Credential names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.303.2 Slots

Class Credential has 8 direct slot definitions:

Uuid

Person

Uid

Provider

Id-Token

Auth-Token

Refresh-Token

Json-Info

## **8.304 Tootsville::Credential-Auth-Token**

### **8.304.1 Function**

Credential-Auth-Token names an undocumented function, with lambda list (INSTANCE).

### **8.304.2 File**

Defined in file src/db/friendly.lisp.

### **8.304.3 SetF Function**

(SETF Credential-Auth-Token) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.304.4 File**

Defined in file src/db/friendly.lisp.

## 8.305 Tootsville::Credential-Id-Token

### 8.305.1 Function

Credential-Id-Token names an undocumented function, with lambda list (INSTANCE).

### 8.305.2 File

Defined in file src/db/friendly.lisp.

### 8.305.3 SetF Function

(SETF Credential-Id-Token) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.305.4 File

Defined in file src/db/friendly.lisp.

## **8.306 Tootsville::Credential-Json-Info**

### **8.306.1 Function**

Credential-Json-Info names an undocumented function, with lambda list (INSTANCE).

### **8.306.2 File**

Defined in file src/db/friendly.lisp.

### **8.306.3 SetF Function**

(SETF Credential-Json-Info) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.306.4 File**

Defined in file src/db/friendly.lisp.

## **8.307 Tootsville::Credential-P**

### **8.307.1 Function**

Credential-P names an undocumented function, with lambda list (OBJECT).

### **8.307.2 File**

Defined in file src/db/friendly.lisp.

## **8.308 Tootsville::Credential-Person**

### **8.308.1 Function**

Credential-Person names an undocumented function, with lambda list (INSTANCE).

### **8.308.2 File**

Defined in file src/db/friendly.lisp.

### **8.308.3 SetF Function**

(SETF Credential-Person) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.308.4 File**

Defined in file src/db/friendly.lisp.



## **8.309 Tootsville::Credential-Provider**

### **8.309.1 Function**

Credential-Provider names an undocumented function, with lambda list (INSTANCE).

### **8.309.2 File**

Defined in file src/db/friendly.lisp.

### **8.309.3 SetF Function**

(SETF Credential-Provider) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.309.4 File**

Defined in file src/db/friendly.lisp.

## **8.310 Tootsville::Credential-Refresh-Token**

### **8.310.1 Function**

Credential-Refresh-Token names an undocumented function, with lambda list (INSTANCE).

### **8.310.2 File**

Defined in file src/db/friendly.lisp.

### **8.310.3 SetF Function**

(SETF Credential-Refresh-Token) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.310.4 File**

Defined in file src/db/friendly.lisp.

## **8.311 Tootsville::Credential-Uid**

### **8.311.1 Function**

Credential-Uid names an undocumented function, with lambda list (INSTANCE).

### **8.311.2 File**

Defined in file src/db/friendly.lisp.

### **8.311.3 SetF Function**

(SETF Credential-Uid) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.311.4 File**

Defined in file src/db/friendly.lisp.

## **8.312 Tootsville::Credential-Uuid**

### **8.312.1 Function**

Credential-Uuid names an undocumented function, with lambda list (INSTANCE).

### **8.312.2 File**

Defined in file src/db/friendly.lisp.

### **8.312.3 SetF Function**

(SETF Credential-Uuid) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.312.4 File**

Defined in file src/db/friendly.lisp.

## 8.313 Tootsville::Cupid-Personality

### 8.313.1 Class

Cupid-Personality names a class, with one superclass: Section 8.1077 [TOOTSVILLE ROBOT-CUPID], page 1362.

This class defines a character named Cupid

### 8.313.2 Slots

Class Cupid-Personality has no direct slots defined.

## **8.314 Tootsville::Current-Position**

### **8.314.1 Function**

Current-Position names an undocumented function, with lambda list (COURSE).

## **8.315 Tootsville::Current-Temp**

### **8.315.1 Function**

Current-Temp names a function, with lambda list (X Y Z):

The current ambient air temperature at X,Y,Z.

### **8.315.2 File**

Defined in file src/weather/weather.lisp.

## 8.316 Tootsville::Database-For

### 8.316.1 Function

Database-For names a function, with lambda list (TYPE):

The database containing the data mirrored by the TYPE

Returns a pairs with the type of database (:MARIA or :COUCH) and the database or schema identification (a proper list).

### 8.316.2 File

Defined in file src/db/generic-db.lisp.



## 8.317 Tootsville::Db-Config

### 8.317.1 Function

Db-Config names an undocumented function, with lambda list (&OPTIONAL (MONIKER \*DB\*)).

### 8.317.2 File

Defined in file src/db/ maria.lisp.

## **8.318 Tootsville::Db-Select**

### **8.318.1 Function**

Db-Select names an undocumented function, with lambda list (QUERY).

### **8.318.2 File**

Defined in file src/db/aria.lisp.

## **8.319 Tootsville::Db-Select-All**

### **8.319.1 Function**

Db-Select-All names an undocumented function, with lambda list (DB QUERY).

### **8.319.2 File**

Defined in file src/db/ maria.lisp.

## 8.320 Tootsville::Db-Select-Records-Simply

### 8.320.1 Function

Db-Select-Records-Simply names a function, with lambda list (TABLE &REST COLUMNS+VALUES):

Query TABLE where columns = values from the plist COLUMNS+VALUES.

Returns all results in a list, so don't use it with a (potentially) large set.

Uses MemCacheD when available.

### 8.320.2 File

Defined in file src/db/maria.lisp.

## 8.321 Tootsville::Db-Select-Single-Column

### 8.321.1 Function

Db-Select-Single-Column names a function, with lambda list (TABLE COLUMN &REST COLUMNS+VALUES):

Select COLUMN from TABLE where columns = values as in plist COLUMNS+VALUES.

Expects to find only one row and return the one column value as an atom.

Signal an error if more rows are returned.

Signals NOT-FOUND if none are found.

Uses MemCacheD when available.

### 8.321.2 File

Defined in file src/db/maria.lisp.

## 8.322 Tootsville::Db-Select-Single-Record

### 8.322.1 Function

Db-Select-Single-Record names a function, with lambda list (TABLE &REST COLUMNS+VALUES):

Select a single record from TABLE where columns = values as in COLUMNS+VALUES.

Calls Section 8.320 [TOOTSVILLE DB-SELECT-RECORDS-SIMPLY], page 574, which in turn may use MemCacheD when it's available.

Signals an error if more than one record is returned.

Signals NOT-FOUND if none are found.

### 8.322.2 File

Defined in file src/db/maria.lisp.

## 8.323 Tootsville::Db-Table-For

### 8.323.1 Function

Db-Table-For names a function, with lambda list (TYPE):

The database table or view containing the data mirrored by the TYPE

### 8.323.2 File

Defined in file src/db/generic-db.lisp.

## **8.324 Tootsville::Debugger**

### **8.324.1 Function**

Debugger names a function, with lambda list NIL:

Start up Swank in the project directory and start a server on the default port.

### **8.324.2 File**

Defined in file src/main.lisp.



## 8.325 Tootsville::Decode-Message

### 8.325.1 Function

Decode-Message names a function, with lambda list (CYPHERTEXT KEY):

Decode the CYPHERTEXT with the KEY.

(FIXME: in what cryptography system?)

### 8.325.2 File

Defined in file `src/endpoints/gossip/alexa/alexa.lisp`.

## **8.326 Tootsville::Decorate-Endpoint-Template-Html**

### **8.326.1 Function**

Decorate-Endpoint-Template-Html names an undocumented function, with lambda list (TEMPLATE VARIABLES METHOD).

### **8.326.2 File**

Defined in file `src/endpoints/slash-meta-game.lisp`.

## 8.327 Tootsville::Decorate-Method-Html

### 8.327.1 Function

Decorate-Method-Html names an undocumented function, with lambda list (METHOD).

### 8.327.2 File

Defined in file `src/endpoints/slash-meta-game.lisp`.

## **8.328 Tootsville::Default-Config-File**

### **8.328.1 Function**

Default-Config-File names a function, with lambda list NIL:

Returns the name of the default configuration file.

### **8.328.2 File**

Defined in file src/config.lisp.

## 8.329 Tootsville::Defendpoint

### 8.329.1 Macro

Defendpoint names a macro, with lambda list ((METHOD URI &OPTIONAL CONTENT-TYPE (HOW-SLOW-IS-SLOW)) &BODY BODY):

Define an HTTP endpoint to access URI via METHOD and return CONTENT-TYPE.

### 8.329.2 File

Defined in file src/web.lisp.

## **8.330 Tootsville::Defendpoint/ Make-Docstring**

### **8.330.1 Function**

Defendpoint/Make-Docstring names an undocumented function, with lambda list (BODY METHOD URI CONTENT-TYPE A-LIST HOW-SLOW-IS-SLOW).

### **8.330.2 File**

Defined in file src/web.lisp.

## 8.331 Tootsville::Defendpoint/ Make-Endpoint-Function

### 8.331.1 Function

Defendpoint/Make-Endpoint-Function names an undocumented function, with lambda list (&KEY FNAME CONTENT-TYPE A-LIST DOCSTRING BODY (HOW-SLOW-IS-SLOW 0.03)).

### 8.331.2 File

Defined in file src/web.lisp.

## **8.332 Tootsville::Define-Alexa-Endpoint**

### **8.332.1 Macro**

Define-Alexa-Endpoint names an undocumented macro, with lambda list (NAME (&OPTIONAL ARG) &BODY BODY).

### **8.332.2 File**

Defined in file `src/endpoints/gossip/alex/alex.lisp`.



### **8.333 Tootsville::Define-Character**

#### **8.333.1 Macro**

Define-Character names an undocumented macro, with lambda list (NAME &OPTIONAL PERSONALITY DOCSTRING).

#### **8.333.2 File**

Defined in file src/characters/characters.lisp.

## 8.334 Tootsville::Define-Maintenance-Task

### 8.334.1 Macro

Define-Maintenance-Task names an undocumented macro, with lambda list (LABEL (NAME START-DELAY FINISH-DELAY) &BODY BODY).

### 8.334.2 File

Defined in file `src/endpoints/slash-maintenance.lisp`.

## 8.335 Tootsville::Define-Operator-Command

### 8.335.1 Macro

Define-Operator-Command names an undocumented macro, with lambda list (COMMAND (WORDS USER PLANE) &BODY BODY).

### 8.335.2 File

Defined in file src/infinity/infinity.lisp.

## **8.336 Tootsville::Define-Personality**

### **8.336.1 Macro**

Define-Personality names an undocumented macro, with lambda list (NAME &OPTIONAL SUPERCLASS).

### **8.336.2 File**

Defined in file src/characters/characters.lisp.

## 8.337 Tootsville::Define-Reply

### 8.337.1 Macro

Define-Reply names an undocumented macro, with lambda list ((LISTENER MODE) &BODY BODY).

### 8.337.2 File

Defined in file src/characters/robots.lisp.

## 8.338 Tootsville::Definfinity

### 8.338.1 Macro

Definfinity names a macro, with lambda list (NAME (LAMBDA-LIST USER-VAR PLANE-VAR) &BODY BODY):

Define an Infinity-mode “c” command NAME.

And now, let’s talk about the Infinity Mode protocol.

### 8.338.2 History of Infinity Mode

In the Beginning, Tootsville used a commercial program called SmartFox Server as its chat server. There were many problems with this, and it didn’t last long.

However, the client program (Persephone) was written to use the SmartFox client libraries, which were very good. So, we kept them, and used an AGPL chat server program created by Bruce-Robert Pocock, named Braque, to replace the server side. Braque was renamed Appius Claudius Cæcus, because SmartFox Server was from Italy, and Appius built the Via Appia, a highway leaving Italy.

Appius gained friends, all of which were given the names of other Romans, and so the entire software suite was nicknamed Romance, like Romans.

In order to negotiate a connection between Appius and SmartFox client, we had to provide a version identifier, so we set the version number to “infinity.”

With the adoption of the gossipnet for Romance II, we had to increment the version from infinity, which brings us to  $\aleph_0$  (read: Alef-Null), which is a particular kind of infinity that is not as big as some other kinds of infinity, as silly as that mathematical construction may sound (yes, that’s real maths).

### 8.338.3 Wire protocols

There are two main wire protocols; RESTful POSTs and gossipnet.

#### 8.338.3.1 RESTful POSTs

The REST POST interface is what you’re really here to read about (on the server side). A POST is submitted with a JSON object representing a command call.

This request can be submitted to either the dedicated endpoint, or the general, dispatching endpoint. The dedicated endpoint will be slightly faster.

The dedicated endpoint will have a URL of the form `/world/infinity/command-name`, with the command name in lower-case and hyphenated.

The dispatching endpoint is `/world/infinity`. Submitting to the dispatching endpoint requires a JSON object with two keys:

- c            The command name, in camelCase
- d            The data to be submitted to that command.

In the case of the dedicated endpoint, only the contents of `d` need to be submitted.

### 8.338.4 Datagram constructions

There are three datagram kinds used in Infinity Mode.

1. The special ‘logOK’ packet type is used only for acknowledging and promulgating login events through the grid. This actually dates back all the way to the SmartFox server’s protocols, so it’s an odd duck.
2. Commands that instigate an action are identified by a ‘c’ attribute.
3. Commands that provide information about the world, usually as a reaction to another event, are called Gatekeeper messages and are identified by a ‘from’ attribute.

### 8.338.5 logOK datagrams

The login process should be documented at Section 8.1346 [TOOTSVILLE WEBSOCKET-AUTHENTICATE], page 1633, for WebSockets, Section 8.1201 [TOOTSVILLE TCP-STREAM-AUTHENTICATE], page 1486, for direct TCP/IP server-to-server streams, and Section 10.243 [Tootsville.Gossip.createConnection], page 2024, for peer-to-peer WebRTC connections.

### 8.338.6 Command datagrams

Command datagrams may be processed through either a REST POST or the Gossipnet. These represent an action or enquiry that a client is making.

Command datagrams are identified by a `c` key, which provides the command name in ‘lowerFirstCamelCase’. This command name is mapped to a function named INFINITY-COMMAND-NAME in hyphenated form.

Command datagrams usually have a `d` key which provides some additional data or parameters to the requested command.

In addition, there may be some of the following. Note that UUID’s are the UUID’s *of a Toot character*, never the person who “owns” that Toot.

`r`

Recipient. This can be an UUID for a direct peer-to-peer command, or is more often just `$World` for the game server or `$All` for all listeners.

`a`

Author. The UUID of the originator of the packet.

`u`

User. The UUID of the user who requested this packet; usually the same as `a/Author`.

`s`

Signature. Proof that the packet originated with `a/Author`.

`v`

Via. The history trail of a forwarded packet.

WRITEME

### 8.338.7 Gatekeeper datagrams

Gatekeeper datagrams are found either as the response to a REST POST, or distributed along the Gossipnet. These represent the state of the world at a certain point in time.

Every Gatekeeper datagram contains the keys `from` and `status`. The `from` value uniquely identifies the type of packet and determines what other fields accompany it. The `status` value is a Boolean, and while its meaning varies by packet, it is usually a good guess that if `status` is not `true`, there has been some kind of request error and data is not available.

For a complete enumeration

### 8.338.8 File

Defined in file `src/infinity/infinity.lisp`.



## 8.339 Tootsville::Defpost

### 8.339.1 Macro

Defpost names a macro, with lambda list (NAME (&KEY) &BODY BODY):

Define a power-on-self-test from somewhere else in the codebase. These are run as confidence tests after a build, or during Production boot-up sequence.

### 8.339.2 File

Defined in file src/power-on-self-test.lisp.

## 8.340 Tootsville::Defrecord

### 8.340.1 Macro

Defrecord names a macro, with lambda list (NAME (DATABASE TABLE &KEY PULL ID-COLUMN) &REST COLUMNS):

Define a database-mapping object type NAME, for DATABASE and TABLE, with COLUMNS.

DATABASE is the symbolic name of the database, mapped via Section 8.248 [TOOTSVILLE CONFIG], page 502; eg, :friendly

TABLE is the string table-name, exactly as it exists in the database; eg, "toots"

PULL is meant to indicate an infrequently-changed, short table (ie, basically a small enumeration) that should be pulled into local cache up-front and referenced from there directly.

COLUMNS are a table of names, types, and foreign-key references, in the form: (LABEL TYPE &rest REFERENCE)

The LABEL of a column is mapped via Section 8.812 [TOOTSVILLE LISP-TO-DB-NAME], page 1097; it is the Lisp name which is essentially the same as the SQL name, but with KEBAB-CASE rather than snake\_case.

When present, REFERENCE is the symbol REF followed by the record-type (class) to whose primary key (ID or UUID) the reference is made. NUMBER REF columns point to ID, UUID REF columns to UUID.

TYPE is one of the following:

NUMBER	map to an integer or real column in the database
STRING	map to a CHAR, CHAR VARYING, or TEXT column, or ENUM
COLOR24	stored in the database as a 24-bit BINARY (3 bytes)
KEYWORD	map to a CHAR or CHAR VARYING column, or ENUM
UUID	stored as a 128-bit BINARY (16 bytes)
JSON	stored as a TEXT column, but parsed on loading via Jonathan
YORNP	a boolean, stored as (typically an enum) 'Y' or 'N'.
URI	stored as CHAR VARYing or TEXT, parsed at load time as a PURI:URI.
TIMESTAMP	translates to a LOCAL-TIME:TIMESTAMP on loading.

### 8.340.2 File

Defined in file src/db/db-central.lisp.

## 8.341 Tootsville::Defrecord/ Before-Save-Normalize

### 8.341.1 Function

Defrecord/Before-Save-Normalize names an undocumented function, with lambda list (NAME COLUMNS).

### 8.341.2 File

Defined in file src/db/db-central.lisp.

## 8.342 Tootsville::Defrecord/ Column-To-Json-Pair

### 8.342.1 Function

Defrecord/Column-To-Json-Pair names an undocumented function, with lambda list (NAME BASENAME COLUMN).

### 8.342.2 File

Defined in file src/db/db-central.lisp.

## 8.343 Tootsville::Defrecord/ Destroy-Record

### 8.343.1 Function

Defrecord/Destroy-Record names an undocumented function, with lambda list (NAME ID-ACCESSOR DATABASE TABLE COLUMNS).

### 8.343.2 File

Defined in file src/db/db-central.lisp.

## 8.344 Tootsville::Defrecord/ Find-Record

### 8.344.1 Function

Defrecord/Find-Record names an undocumented function, with lambda list (NAME TABLE COLUMNS).

### 8.344.2 File

Defined in file src/db/db-central.lisp.

## 8.345 Tootsville::Defrecord/ Find-Record/ Pull

### 8.345.1 Function

Defrecord/Find-Record/Pull names an undocumented function, with lambda list (NAME TABLE COLUMNS).

### 8.345.2 File

Defined in file src/db/db-central.lisp.

## **8.346 Tootsville::Defrecord/ Find-Records**

### **8.346.1 Function**

Defrecord/Find-Records names an undocumented function, with lambda list (NAME TABLE COLUMNS).

### **8.346.2 File**

Defined in file src/db/db-central.lisp.



## 8.347 Tootsville::Defrecord/ Find-Records-By-Sql

### 8.347.1 Function

Defrecord/Find-Records-By-Sql names an undocumented function, with lambda list (NAME DATABASE).

### 8.347.2 File

Defined in file src/db/db-central.lisp.

## **8.348 Tootsville::Defrecord/ Find-Records/ Pull**

### **8.348.1 Function**

Defrecord/Find-Records/Pull names an undocumented function, with lambda list (NAME TABLE COLUMNS).

### **8.348.2 File**

Defined in file src/db/db-central.lisp.

## 8.349 Tootsville::Defrecord/ Find-Reference

### 8.349.1 Function

Defrecord/Find-Reference names an undocumented function, with lambda list (NAME COLUMN).

### 8.349.2 File

Defined in file src/db/db-central.lisp.

## **8.350 Tootsville::Defrecord/ Find-Reference-Columns**

### **8.350.1 Function**

Defrecord/Find-Reference-Columns names an undocumented function, with lambda list (NAME COLUMNS).

### **8.350.2 File**

Defined in file src/db/db-central.lisp.

## 8.351 Tootsville::Defrecord/ Id-Column-For

### 8.351.1 Function

Defrecord/Id-Column-For names an undocumented function, with lambda list (NAME COLUMNS ID-COLUMN).

### 8.351.2 File

Defined in file src/db/db-central.lisp.

## 8.352 Tootsville::Defrecord/ Invalidate-Cache

### 8.352.1 Function

Defrecord/Invalidate-Cache names an undocumented function, with lambda list (NAME PULL COLUMNS).

### 8.352.2 File

Defined in file src/db/db-central.lisp.

## 8.353 Tootsville::Defrecord/ Load-Record

### 8.353.1 Function

Defrecord/Load-Record names an undocumented function, with lambda list (NAME COLUMNS).

### 8.353.2 File

Defined in file src/db/db-central.lisp.

## **8.354 Tootsville::Defrecord/ Make-Record**

### **8.354.1 Function**

Defrecord/Make-Record names an undocumented function, with lambda list (NAME).

### **8.354.2 File**

Defined in file src/db/db-central.lisp.



## 8.355 Tootsville::Defrecord/ Record=

### 8.355.1 Function

Defrecord/Record= names an undocumented function, with lambda list (NAME ID-ACCESSOR).

### 8.355.2 File

Defined in file src/db/db-central.lisp.

## **8.356 Tootsville::Defrecord/ Save-Record**

### **8.356.1 Function**

Defrecord/Save-Record names an undocumented function, with lambda list (NAME ID-ACCESSOR DATABASE TABLE COLUMNS).

### **8.356.2 File**

Defined in file src/db/db-central.lisp.

## 8.357 Tootsville::Defrecord/ Save-Record-With-Id-Column

### 8.357.1 Function

Defrecord/Save-Record-With-Id-Column names an undocumented function, with lambda list (NAME DATABASE TABLE COLUMNS).

### 8.357.2 File

Defined in file src/db/db-central.lisp.

## **8.358 Tootsville::Defrecord/ To-Json**

### **8.358.1 Function**

Defrecord/To-Json names an undocumented function, with lambda list (NAME COLUMNS).

### **8.358.2 File**

Defined in file src/db/db-central.lisp.

## **8.359 Tootsville::Delete-Contact**

### **8.359.1 Function**

Delete-Contact names an undocumented function, with lambda list (OWNER CONTACT).

### **8.359.2 File**

Defined in file src/contacts.lisp.

## 8.360 Tootsville::Demand-Quiesce-Toot

### 8.360.1 Function

Demand-Quiesce-Toot names a function, with lambda list (TOOT):

Send TOOT a demand that it quiesce a copy of itself to the database

### 8.360.2 File

Defined in file src/toots.lisp.

## 8.361 Tootsville::Describe-System

### 8.361.1 Function

Describe-System names a function, with lambda list (SYSTEM S):

Describes an ASDF system SYSTEM to stream S.

This is used to obtain the title, description, author, maintainer, and license information, as well as the contents of any COPYING or LICENSE file, in TeXinfo format for inclusion in the manual.

### 8.361.2 Example Output

```
@subsection System Tootsville
```

```
The server software monolith for REST services of Tootsville.org
```

```
Author: Bruce-Robert Pocock <BRPocock@@ciwta.org>
```

```
License: AGPL v3+
```

### 8.361.3 File

Defined in file src/main.lisp.

## 8.362 Tootsville::Describe-World

### 8.362.1 Function

Describe-World names a function, with lambda list (LATITUDE LONGITUDE ALTITUDE WORLD):

Describe the world at LATITUDE, LONGITUDE, ALTITUDE in WORLD.

Returns a PList with :TERRAIN and :FURNITURE.

:TERRAIN contains a 201×201 grid of 1m corners of a 200×200 meter space (that is, the entire space at LATITUDE, LONGITUDE at ALTITUDE=0). If ALTITUDE is not zero, the :TERRAIN is omitted.

:FURNITURE contains a list of item descriptions as per Section 8.799 [TOOTSVILLE ITEMS-AT], page 1084, which are as per Section 8.770 [TOOTSVILLE ITEM-INFO], page 1055.

### 8.362.2 File

Defined in file src/world.lisp.



## **8.363 Tootsville::Destroy-All-Idle-Workers**

### **8.363.1 Function**

Destroy-All-Idle-Workers names a function, with lambda list NIL:

Destroy all idle web worker threads violently.

Normally only needed during debugging. See instead Section 8.1167 [TOOTSVILLE STOP], page 1452.

### **8.363.2 File**

Defined in file src/main.lisp.

## **8.364 Tootsville::Destroy-All-Listeners**

### **8.364.1 Function**

Destroy-All-Listeners names a function, with lambda list NIL:

Destroy all Hunchentoot listener threads violently.

Normally only needed during debugging. See instead Section 8.1167 [TOOTSVILLE STOP], page 1452.

### **8.364.2 File**

Defined in file src/main.lisp.

## **8.365 Tootsville::Destroy-All-Web-Tasks**

### **8.365.1 Function**

Destroy-All-Web-Tasks names a function, with lambda list NIL:

Destroy all web listeners and worker threads.

May make a second (or subsequent) pass to try to clean up non-idle worker threads after 1 second, but no guarantee that it will destroy them all.

### **8.365.2 File**

Defined in file src/main.lisp.

## **8.366 Tootsville::Destroy-Endpoint**

### **8.366.1 Function**

Destroy-Endpoint names an undocumented function, with lambda list (METHOD URI &OPTIONAL CONTENT-TYPE).

### **8.366.2 File**

Defined in file src/web.lisp.

## 8.367 Tootsville::Destroy-Record

### 8.367.1 Function

Destroy-Record names a function, with lambda list (OBJECT):

Delete the record in the database representing OBJECT.

Does not attempt to destroy OBJECT, so a subsequent call to Section 8.1115 [TOOTSVILLE SAVE-RECORD], page 1400, could potentially be used to re-create it.

### 8.367.2 File

Defined in file src/db/generic-db.lisp.

## **8.368 Tootsville::Destroy-Toot**

### **8.368.1 Function**

Destroy-Toot names a function, with lambda list (TOOT):

Prompt (with CERROR (see the Common Lisp HyperSpec)) to Section 8.367 [TOOTSVILLE DESTROY-RECORD], page 623, TOOT

### **8.368.2 File**

Defined in file src/toots.lisp.

## 8.369 Tootsville::Detailed-Time

### 8.369.1 Function

Detailed-Time names a function, with lambda list (&OPTIONAL (NOW (GET-UNIVERSAL-TIME))):

Get a long string explaining the date, time, and other info.

This is a multiple-paragraph explanation of the form:

```
Currently the universal time code is 3,783,302,735.
```

```
On the planet Chœorgryllum, it is 05:25:35 in the wee hours of the
morning on Blanksday, the fifth of Inunguis, 152.
```

```
That's the ninth day of the nine-day week, and the third month of the
twelve months of the year.
```

```
It is the fifth day of The Moon's 30-day month.
```

```
It is the sixty-fourth day of The Other Moon's 71-day month.
```

```
It is the thirty-seventh day of The Pink Moon's 53-day month.
```

```
It is the 95th day of the 360-day calendar year.
```

If the day is an holiday, it will also be mentioned.

### 8.369.2 File

Defined in file src/endpoints/slash-world.lisp.

## **8.370 Tootsville::Devel**

### **8.370.1 Variable**

Devel names an undocumented variable with the value NIL



## **8.371 Tootsville::Disable-Sbcl-Ldb**

### **8.371.1 Function**

Disable-Sbcl-Ldb names an undocumented function, with lambda list NIL.

### **8.371.2 File**

Defined in file src/utls.lisp.

## 8.372 Tootsville::Disconnect-No-Login

### 8.372.1 Function

Disconnect-No-Login names a function, with lambda list (CLIENT):

Disconnect client for failing to log in within the required time or number of commands.

### 8.372.2 File

Defined in file src/websockets.lisp.

## **8.373 Tootsville::Dispatch-Endpoint**

### **8.373.1 Function**

Dispatch-Endpoint names an undocumented function, with lambda list (MATCH).

### **8.373.2 File**

Defined in file src/acceptor.lisp.

## **8.374 Tootsville::Distance**

### **8.374.1 Function**

Distance names an undocumented function, with lambda list (X1 Y1 Z1 X2 Y2 Z2).

### **8.374.2 File**

Defined in file src/world.lisp.

## **8.375 Tootsville::Divisible-By-200-P**

### **8.375.1 Function**

Divisible-By-200-P names a function, with lambda list (N):

Is N evenly divisible by 200?

### **8.375.2 File**

Defined in file src/utils.lisp.

## **8.376 Tootsville::Dns-Name**

### **8.376.1 Type**

Dns-Name names a TYPE:

A name that could be a DNS hostname

See Section 8.665 [TOOTSVILLE HOST-NAME-LIKE-P], page 923.

## 8.377 Tootsville::Do-After

### 8.377.1 Macro

Do-After names a macro, with lambda list ((TIME) &BODY BODY):

Perform BODY after TIME seconds have elapsed.

Uses a one-shot-timer metronome facility.

### 8.377.2 File

Defined in file src/metronome.lisp.

## 8.378 Tootsville::Do-Db-Records-Simply

### 8.378.1 Macro

Do-Db-Records-Simply names a macro, with lambda list ((RECORD-VAR TABLE &REST COLUMNS+VALUES) &BODY BODY):

Iterate RECORD-VAR over TABLE where columns = values as in plist COLUMNS+VALUES.

Selects one record at a time from TABLE. Does not use MemCacheD.

### 8.378.2 File

Defined in file src/db/maria.lisp.



## 8.379 Tootsville::Do-Metronome

### 8.379.1 Macro

Do-Metronome names a macro, with lambda list `((&KEY FREQUENCY ONE-SHOT-TIME NAME) &BODY BODY)`:

Perform `BODY` as a metronome facility named `NAME`, at `FREQUENCY` or once at `ONE-SHOT-TIME`.

`FREQUENCY` is given in seconds, or `ONE-SHOT-TIME` is given in Universal time. When both are given, the facility will execute at the rate of `FREQUENCY` until a final execution at `ONE-SHOT-TIME`.

### 8.379.2 File

Defined in file `src/metronome.lisp`.

## 8.380 Tootsville::Do-Records

### 8.380.1 Macro

Do-Records names a macro, with lambda list ((RECORD-VAR TYPE &REST COLUMNS+VALUES) &BODY BODY):

Apply BODY to each row as if Section 8.576 [TOOTSVILLE FIND-RECORDS], page 833, were called.

### 8.380.2 File

Defined in file src/db/db-central.lisp.

## **8.381 Tootsville::Docstring->Html**

### **8.381.1 Function**

Docstring->Html names an undocumented function, with lambda list (DOCSTRING SYMBOL).

### **8.381.2 File**

Defined in file `src/endpoints/slash-meta-game.lisp`.

## **8.382 Tootsville::Docstring->Markdown**

### **8.382.1 Function**

Docstring->Markdown names an undocumented function, with lambda list (DOCSTRING).

### **8.382.2 File**

Defined in file `src/endpoints/slash-meta-game.lisp`.

## **8.383 Tootsville::Doff-Any-Conflicting-Item**

### **8.383.1 Function**

Doff-Any-Conflicting-Item names an undocumented function, with lambda list (WEAR-SLOT TOOT).

### **8.383.2 File**

Defined in file src/items.lisp.

## **8.384 Tootsville::Doff-Item**

### **8.384.1 Function**

Doff-Item names a function, with lambda list (INVENTORY-ITEM):

Un-equip ITEM.

### **8.384.2 File**

Defined in file src/items.lisp.

## **8.385 Tootsville::Doff-Item-In-Slot**

### **8.385.1 Function**

Doff-Item-In-Slot names an undocumented function, with lambda list (WEAR-SLOT TOOT).

### **8.385.2 File**

Defined in file src/items.lisp.

## 8.386 Tootsville::Don-Item

### 8.386.1 Function

Don-Item names a function, with lambda list (INVENTORY-ITEM WEAR-SLOT):

Equip INVENTORY-ITEM on its owning Toot in SLOT.

If this conflicts with any other equipped items, remove them.

### 8.386.2 File

Defined in file src/items.lisp.



## **8.387 Tootsville::Doodle-Personality**

### **8.387.1 Class**

Doodle-Personality names a class, with one superclass: Section 8.1078 [TOOTSVILLE ROBOT-DOODLE], page 1363.

This class defines a character named Doodle

### **8.387.2 Slots**

Class Doodle-Personality has no direct slots defined.

## **8.388 Tootsville::Dottie-Personality**

### **8.388.1 Class**

Dottie-Personality names a class, with one superclass: Section 8.1079 [TOOTSVILLE ROBOT-DOTTIE], page 1364.

This class defines a character named Dottie

### **8.388.2 Slots**

Class Dottie-Personality has no direct slots defined.

## 8.389 Tootsville::Double-@

### 8.389.1 Function

Double-@ names a function, with lambda list (STRING):

    Edit STRING to be safe in TeXInfo.

    Escapes @, {, and } characters and adds a space after “/” characters.

### 8.389.2 File

Defined in file src/write-docs-2.lisp.

## **8.390 Tootsville::Drop-Item**

### **8.390.1 Function**

Drop-Item names a function, with lambda list (INVENTORY-ITEM):

Drop ITEM and cease to own it.

### **8.390.2 File**

Defined in file src/items.lisp.

## 8.391 Tootsville::Dump-Credits

### 8.391.1 Function

Dump-Credits names a function, with lambda list NIL:

Send +CREDITS+ as a private admin message. Response to the ,credits user utterance.

### 8.391.2 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.392 Tootsville::Dump-Global-Heightmap**

### **8.392.1 Function**

Dump-Global-Heightmap names an undocumented function, with lambda list (LATITUDE LONGITUDE).

### **8.392.2 File**

Defined in file src/terrain.lisp.

## **8.393 Tootsville::Email-Lhs**

### **8.393.1 Function**

Email-Lhs names an undocumented function, with lambda list (ADDRESS).

### **8.393.2 File**

Defined in file src/users.lisp.

## **8.394 Tootsville::Enable-Sbcl-Ldb**

### **8.394.1 Function**

Enable-Sbcl-Ldb names an undocumented function, with lambda list NIL.

### **8.394.2 File**

Defined in file src/utls.lisp.



## **8.395 Tootsville::Enable-Ssl-P**

### **8.395.1 Function**

Enable-Ssl-P names an undocumented function, with lambda list NIL.

### **8.395.2 File**

Defined in file src/config.lisp.

## 8.396 Tootsville::Encode-Endpoint-Reply

### 8.396.1 Function

Encode-Endpoint-Reply names a function, with lambda list (REPLY):

Handle the reply from an endpoint function gracefully.

Strings are sent in UTF-8.

Vectors are assumed to be octet vectors.

Lists can begin with a status code number, followed by an optional list of headers, followed by actual contents. A list not beginning with a status number is assumed to be cons data, which is transmitted as JSON in UTF-8 using the Jonathan transcoding.

Relies upon Section 8.265 [TOOTSVILLE CONTENTS-TO-BYTES], page 519, qv

### 8.396.2 File

Defined in file src/web.lisp.

## 8.397 Tootsville::Endpoint

### 8.397.1 Class

Endpoint names a class, with one superclass: COMMON-LISP::STANDARD-OBJECT (not in this manual).

### 8.397.2 Slots

Class Endpoint has 6 direct slot definitions:

**Function** (undocumented)

**Method** (undocumented)

**Template** (undocumented)

**Template-Arity**  
(undocumented)

**Content-Type**  
(undocumented)

**How-Slow-Is-Slow**  
(undocumented)

## **8.398 Tootsville::Endpoint->Html**

### **8.398.1 Function**

Endpoint->Html names an undocumented function, with lambda list (ENDPOINT).

### **8.398.2 File**

Defined in file `src/endpoints/slash-meta-game.lisp`.

## **8.399 Tootsville::Endpoint->Openapi**

### **8.399.1 Function**

Endpoint->Openapi names a function, with lambda list (ENDPOINT):

Convert an ENDPOINT description PList into an OpenAPI description.

### **8.399.2 File**

Defined in file src/endpoints/slash-meta-game.lisp.

## **8.400 Tootsville::Endpoint->Plist**

### **8.400.1 Function**

Endpoint->Plist names an undocumented function, with lambda list (ENDPOINT).

### **8.400.2 File**

Defined in file `src/endpoints/slash-meta-game.lisp`.

## 8.401 Tootsville::Endpoint-Close

### 8.401.1 Function

Endpoint-Close names a function, with lambda list (ENDPOINT METHOD ARITY UA-ACCEPT):

Is the given ENDPOINT similar to METHOD ARITY UA-ACCEPT?

This is used to quickly filter endpoints using only fast integer = (see the Common Lisp HyperSpec) and symbol EQL (see the Common Lisp HyperSpec) comparisons, so that the more expensive template unification algorithm can run only on fewer, relatively similar URIs.

### 8.401.2 File

Defined in file src/endpoint.lisp.

## **8.402 Tootsville::Endpoint-Close-Key**

### **8.402.1 Function**

Endpoint-Close-Key names a function, with lambda list (ENDPOINT):

A small list that acts like a hash for ENDPOINT. Serves the same purpose as Section 8.401 [TOOTSVILLE ENDPOINT-CLOSE], page 657.

### **8.402.2 File**

Defined in file src/endpoint.lisp.



## **8.403 Tootsville::Endpoint-Content-Type**

### **8.403.1 Function**

Endpoint-Content-Type names an undocumented function, with lambda list (OBJECT).

## 8.404 Tootsville::Endpoint-Delete-/ Users/ Me/ Toots/ Toot-Name $\mapsto$ Json

### 8.404.1 Function

Endpoint-Delete-/ Users/ Me/ Toots/ Toot-Name $\mapsto$ Json names a function, with lambda list (TOOT-NAME):

Permanently destroys the Toot character TOOT-NAME.

This Toot must be owned by you (the logged-in user).

Any inventory held by the Toot, or property owned by the Toot, will be released to the public domain. Players should transfer items or property prior to deleting a Toot.

For a time after a Toot's deletion, their name remains locked (to prevent immediate impersonation).

Requires player authentication.

### 8.404.2 Status: 202 Toot deletion in progress

The Toot will be deleted, but it may not have completed yet. A subsequent, identical request can confirm.

### 8.404.3 Status: 204 Toot deleted

The Toot has been deleted. Repeated calls will return the same status, for the duration of the name lock on the Toot.

### 8.404.4 Status: 401 Authorization Required

No user credentials were passed.

### 8.404.5 Status: 403 Authorization Failed

The user credentials presented were not recognized.

### 8.404.6 Status: 404 Not Found

The Toot named does not exist.

### 8.404.7 Status: 405 Not Allowed

The Toot named is one that you have permission to use, but are not the main owner of. This is usually a child account.

### 8.404.8 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method DELETE at the URI template /users/me/toots/:toot-name. It returns a content-type of application/json.

TOOT-NAME is a parameter from the URI.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.404.9 File

Defined in file src/endpoints/slash-users.lisp.

## **8.405 Tootsville::Endpoint-Function**

### **8.405.1 Function**

Endpoint-Function names an undocumented function, with lambda list (OBJECT).

## 8.406 Tootsville::Endpoint-Get-/ Favicon/ Ico→Vnd.Microsoft.Icon

### 8.406.1 Function

Endpoint-Get-/ Favicon/ Ico→Vnd.Microsoft.Icon names a function, with lambda list NIL:

Get the Tootsville logo in Windows Icon format

### 8.406.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /favicon/ico. It returns a content-type of image/vnd.microsoft.icon.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.406.3 File

Defined in file src/web.lisp.

## 8.407 Tootsville::Endpoint-Get-/ Favicon→Gif

### 8.407.1 Function

Endpoint-Get-/ Favicon→Gif names a function, with lambda list NIL:

Get the Tootsville logo as a GIF

### 8.407.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /favicon. It returns a content-type of image/gif.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.407.3 File

Defined in file src/web.lisp.

## 8.408 Tootsville::Endpoint-Get-/ Favicon→Png

### 8.408.1 Function

Endpoint-Get-/ Favicon→Png names a function, with lambda list NIL:

Get the Tootsville logo as a PNG

### 8.408.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /favicon. It returns a content-type of image/png.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.408.3 File

Defined in file src/web.lisp.

## 8.409 Tootsville::Endpoint-Get-/ Gossip/ Answers/ Uuid $\mapsto$ Sdp

### 8.409.1 Function

Endpoint-Get-/ Gossip/ Answers/ Uuid $\mapsto$ Sdp names a function, with lambda list (UUID):

Read back the answer to an offer posted previously.

This is a COMET-type call which may sleep up to 30s.

### 8.409.2 204 No Content

No Content is returned if the offer has not yet been accepted. However, this will not be returned immediately; the host will wait up to 30s before returning failure.

### 8.409.3 200 OK

The SDP answer will be returned.

### 8.409.4 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /gossip/answers/:uuid. It returns a content-type of application/sdp.

UUID is a parameter from the URI.

It will report a slow response if it takes longer than 31.0 seconds (31,000 milliseconds) to complete.

### 8.409.5 File

Defined in file src/endpoints/slash-gossip.lisp.

## **8.410 Tootsville::Endpoint-Get-/ Gossip/ Ice-Servers→Json**

### **8.410.1 Function**

Endpoint-Get-/ Gossip/ Ice-Servers→Json names a function, with lambda list NIL:

Obtain STUN/TURN server credentials for ICE.

### **8.410.2 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method GET at the URI template /gossip/ice-servers. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.410.3 File**

Defined in file src/endpoints/slash-gossip.lisp.



## 8.411 Tootsville::Endpoint-Get-/ Gossip/ Offers $\mapsto$ Json

### 8.411.1 Function

Endpoint-Get-/ Gossip/ Offers $\mapsto$ Json names a function, with lambda list NIL:

Ask for any, arbitrary offer to potentially accept.

Returns a JSON object with UUID (for answering) and SDP description.

### 8.411.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /gossip/offers. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.411.3 File

Defined in file src/endpoints/slash-gossip.lisp.

## 8.412 Tootsville::Endpoint-Get-/ Maintenance/ ↦Txt

### 8.412.1 Function

Endpoint-Get-/ Maintenance/ ↦Txt names a function, with lambda list NIL:

Undocumented endpoint for GET /maintenance/ ↦ :TEXT/PLAIN

### 8.412.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /maintenance/. It returns a content-type of text/plain.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.412.3 File

Defined in file src/endpoints/slash-maintenance.lisp.

## 8.413 Tootsville::Endpoint-Get-/ Meta-Game/ Headers $\mapsto$ Json

### 8.413.1 Function

Endpoint-Get-/ Meta-Game/ Headers $\mapsto$ Json names a function, with lambda list NIL:

This method returns to the user, the headers that reached the application server.

Note that these may have been modified by proxies or load-balancers in transit.

### 8.413.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /meta-game/headers. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.413.3 File

Defined in file src/endpoints/slash-meta-game.lisp.

## 8.414 Tootsville::Endpoint-Get-/ Meta-Game/ Ping $\mapsto$ Txt

### 8.414.1 Function

Endpoint-Get-/ Meta-Game/ Ping $\mapsto$ Txt names a function, with lambda list NIL:

Ping the server (test server presence and latency)

### 8.414.2 200: OK

This endpoint always returns the 6-character string: “pong”

### 8.414.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /meta-game/ping. It returns a content-type of text/plain.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.414.4 File

Defined in file src/endpoints/slash-meta-game.lisp.

## 8.415 Tootsville::Endpoint-Get-/ Meta-Game/ Services/ Old→Json

### 8.415.1 Function

Endpoint-Get-/ Meta-Game/ Services/ Old→Json names a function, with lambda list NIL:

This is a sketchy sort of listing of services in a JSON format that is not anybody's standard. It exists as a stop-gap measure until the OpenAPI form is working nicely.

### 8.415.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /meta-game/services/old. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.415.3 File

Defined in file src/endpoints/slash-meta-game.lisp.

## 8.416 Tootsville::Endpoint-Get-/ Meta-Game/ Services $\mapsto$ Html

### 8.416.1 Function

Endpoint-Get-/ Meta-Game/ Services $\mapsto$ Html names a function, with lambda list NIL:

Provide a listing of services available in this cluster.

This provides a browseable catalog of web services that are provided by this machine or its siblings.

### 8.416.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /meta-game/services. It returns a content-type of text/html.

There are no URI parameters.

It will report a slow response if it takes longer than 0.15 seconds (150 milliseconds) to complete.

### 8.416.3 File

Defined in file src/endpoints/slash-meta-game.lisp.

## 8.417 Tootsville::Endpoint-Get-/ Meta-Game/ Services $\mapsto$ Json

### 8.417.1 Function

Endpoint-Get-/ Meta-Game/ Services $\mapsto$ Json names a function, with lambda list NIL:

Enumerate services for OpenAPI.

Provide an OpenAPI JSON dump of the same information seen on this page, but in a machine-readable format.

### 8.417.2 Status: 200 OK

The data returned is in the JSON encoded form of OpenAPI 3.0.0; see <https://openapis.org/> for details.

### 8.417.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /meta-game/services. It returns a content-type of application/vnd.oai.openapi;version=3.0.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.417.4 File

Defined in file src/endpoints/slash-meta-game.lisp.

## 8.418 Tootsville::Endpoint-Get-/ Toots/ Toot-Name→Json

### 8.418.1 Function

Endpoint-Get-/ Toots/ Toot-Name→Json names a function, with lambda list (TOOT-NAME):

Get public info about TOOT-NAME

TOOT-NAME is the name of a Toot or other character-type avatar in the game. This endpoint will return the public information about TOOT-NAME in the format described by Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534.

### 8.418.2 200 OK

Note that the HTTP Last-Modified header will be set to the Section 8.1252 [TOOTSVILLE TOOT-LAST-ACTIVE], page 1539, time of the Toot character.

The body of the response will be the public information about TOOT-NAME in the form described at Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534.

### 8.418.3 404 Not Found

This is returned if TOOT-NAME does not name a character in the game.

### 8.418.4 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /toots/:toot-name. It returns a content-type of application/json.

TOOT-NAME is a parameter from the URI.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.418.5 File

Defined in file src/endpoints/slash-toots.lisp.



## 8.419 Tootsville::Endpoint-Get-/ Toots/ Toot-Name $\mapsto$ Txt

### 8.419.1 Function

Endpoint-Get-/ Toots/ Toot-Name $\mapsto$ Txt names a function, with lambda list (TOOT-NAME):

Get public info about TOOT-NAME

### 8.419.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /toots/:toot-name. It returns a content-type of text/plain.

TOOT-NAME is a parameter from the URI.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.419.3 File

Defined in file src/endpoints/slash-toots.lisp.

## 8.420 Tootsville::Endpoint-Get-/ Users/ Me/ Toots/ Toot-Name→Json

### 8.420.1 Function

Endpoint-Get-/ Users/ Me/ Toots/ Toot-Name→Json names a function, with lambda list (TOOT-NAME):

Gives detailed information about your Toot character TOOT-NAME.

This Toot must be owned by you (the logged-in user). You will receive details about your own Toot, like inventory, that are not available to other players.

Requires player authentication.

### 8.420.2 Status: 200 OK

Returns the JSON information as per Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534, including private information that is only available to the owner.

Note that the `Last-Modified` header will be set to the Toot's Section 8.1252 [TOOTSVILLE TOOT-LAST-ACTIVE], page 1539, value.

### 8.420.3 Status: 401 Authorization Required

No user credentials were passed.

### 8.420.4 Status: 403 Authorization Failed

The user credentials presented were not recognized, or the Toot name given is not your Toot.

### 8.420.5 Status: 404 Not Found

The Toot name given was not registered.

### 8.420.6 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template `/users/me/toots/:toot-name`. It returns a content-type of `application/json`.

TOOT-NAME is a parameter from the URI.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.420.7 File

Defined in file `src/endpoints/slash-users.lisp`.

## 8.421 Tootsville::Endpoint-Get-/ Users/ Me/ Toots/ Toot-Name $\mapsto$ Txt

### 8.421.1 Function

Endpoint-Get-/ Users/ Me/ Toots/ Toot-Name $\mapsto$ Txt names a function, with lambda list (TOOT-NAME):

Gives detailed information about your Toot character TOOT-NAME.

This text information is an English-like rendering of the same information as Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534, returns.

### 8.421.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /users/me/toots/:toot-name. It returns a content-type of text/plain.

TOOT-NAME is a parameter from the URI.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.421.3 File

Defined in file src/endpoints/slash-users.lisp.

## 8.422 Tootsville::Endpoint-Get-/ Users/ Me/ Toots $\mapsto$ Json

### 8.422.1 Function

Endpoint-Get-/ Users/ Me/ Toots $\mapsto$ Json names a function, with lambda list NIL:

Enumerate all Toot characters available to you.

### 8.422.2 200 OK

The `Last-Modified` header will be set to the maximum Section 8.1252 [TOOTSVILLE TOOT-LAST-ACTIVE], page 1539, time of any Toot in roster.

The returned body will have a JSON object with one key, `toots`, which in turn contains an array of Toot names, ordered by their Section 8.1252 [TOOTSVILLE TOOT-LAST-ACTIVE], page 1539, time, with the most recently used Toot listed first.

### 8.422.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template `/users/me/toots`. It returns a content-type of `application/json`.

There are no URI parameters.

It will report a slow response if it takes longer than 1.0 seconds (1,000 milliseconds) to complete.

### 8.422.4 File

Defined in file `src/endpoints/slash-users.lisp`.

## 8.423 Tootsville::Endpoint-Get-/ Users/ Me/ Toots $\mapsto$ Txt

### 8.423.1 Function

Endpoint-Get-/ Users/ Me/ Toots $\mapsto$ Txt names a function, with lambda list NIL:

Enumerate all Toot characters available to you.

### 8.423.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /users/me/toots. It returns a content-type of text/plain.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.423.3 File

Defined in file src/endpoints/slash-users.lisp.

## 8.424 Tootsville::Endpoint-Get-/ Users/ Me $\mapsto$ Json

### 8.424.1 Function

Endpoint-Get-/ Users/ Me $\mapsto$ Json names a function, with lambda list NIL:

Provides information about your user account.

Requires player authentication.

### 8.424.2 Status: 200 OK

You will receive some information about your user account.

The top-level keys of the JSON object are:

TODO: document this properly

### 8.424.3 Status: 401 Authorization Required

### 8.424.4 Status: 403 Authorization Failed

### 8.424.5 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /users/me. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.424.6 File

Defined in file src/endpoints/slash-users.lisp.

## 8.425 Tootsville::Endpoint-Get-/ Users/ Me→Txt

### 8.425.1 Function

Endpoint-Get-/ Users/ Me→Txt names a function, with lambda list NIL:

Provides information about your user account, in plain text form.

### 8.425.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /users/me. It returns a content-type of text/plain.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.425.3 File

Defined in file src/endpoints/slash-users.lisp.

## 8.426 Tootsville::Endpoint-Get-/ Version/ About/ Detail/ Param $\mapsto$ Json

### 8.426.1 Function

Endpoint-Get-/ Version/ About/ Detail/ Param $\mapsto$ Json names a function, with lambda list (PARAM):

Returns (as a JSON object) the info specified by PARAM.

See the endpoint GET /version/about/detail/:param.txt for a list of possible values of PARAM.

### 8.426.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /version/about/detail/:param. It returns a content-type of application/json.

PARAM is a parameter from the URI.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.426.3 File

Defined in file src/endpoints/slash-version.lisp.



## 8.427 Tootsville::Endpoint-Get-/ Version/ About/ Detail/ Param $\mapsto$ Txt

### 8.427.1 Function

Endpoint-Get-/ Version/ About/ Detail/ Param $\mapsto$ Txt names a function, with lambda list (PARAM):

Returns (as plain text) the info specified by PARAM.

The values available can be seen by GET /version/about.txt, but include the following. Values are case-insensitive.

**Product** The product running (eg, Tootsville)

**Version** The current version of the application

**Copyright**  
The copyright notice (one-line form) for the application

**Environment/Configuration**  
The environment configuration being run within

**Environment/Developmentp**  
True if this is a development server

**Environment/Productionp**  
True if this is a production server

**Machine/Version**  
The MACHINE-VERSION (see the Common Lisp HyperSpec) on which this is running.

**Machine/Type**  
The MACHINE-TYPE (see the Common Lisp HyperSpec) on which this is running.

**Machine/Instance**  
The MACHINE-INSTANCE (see the Common Lisp HyperSpec) on which this is running.

**Site/Short-Name**  
The short name of the active site.

**Site/Long-Name**  
The long name of the active site.

**Lisp/Type**  
The 'LISP-TYPE' of the compiler used.

**Lisp/Version**  
The 'LISP-VERSION' of the compiler used.

**Software/Type**  
The SOFTWARE-TYPE (see the Common Lisp HyperSpec) of the operating system.

**Software/Version**  
The SOFTWARE-VERSION (see the Common Lisp HyperSpec) of the operating system.

**Copyright-Latest**

The latest year in the copyright declaration.

**Build-Date**

The date on which the software was first compiled.

**Compiled** The point in time at which the software was compiled.

**Request/Name**

The requestor name

**Request/Port**

The port on which the request was made

**Request/Protocol**

The protocol via which the request was made

**Acceptor/Class**

The class of the acceptor handling this request

**Acceptor/Name**

The name of the acceptor handling this request

**Acceptor/Port**

The port on which the acceptor is handling this request

**Acceptor/Address**

The address on which the acceptor is handling this request

### **8.427.2 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method GET at the URI template `/version/about/detail/:param`. It returns a content-type of `text/plain`.

PARAM is a parameter from the URI.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.427.3 File**

Defined in file `src/endpoints/slash-version.lisp`.

## 8.428 Tootsville::Endpoint-Get-/ Version/ About $\mapsto$ Json

### 8.428.1 Function

Endpoint-Get-/ Version/ About $\mapsto$ Json names a function, with lambda list NIL:

Returns all version information about this host.

### 8.428.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /version/about. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.428.3 File

Defined in file src/endpoints/slash-version.lisp.

## 8.429 Tootsville::Endpoint-Get-/ Version/ About $\mapsto$ Txt

### 8.429.1 Function

Endpoint-Get-/ Version/ About $\mapsto$ Txt names a function, with lambda list NIL:

Returns all version information about this host.

### 8.429.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /version/about. It returns a content-type of text/plain.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.429.3 File

Defined in file src/endpoints/slash-version.lisp.

## 8.430 Tootsville::Endpoint-Get-/ World/ Clock/ Calendar/ Now/ Fragment $\mapsto$ Html

### 8.430.1 Function

Endpoint-Get-/ World/ Clock/ Calendar/ Now/ Fragment $\mapsto$ Html names a function, with lambda list NIL:

Get a calendar fragment in HTML for the current month of the current year.

The HTML fragment contains the month header, day-of-week headers, and has holidays marked with HTML title attributes.

See Section 3.4 [CHCEROGRYLLUM CAL-MONTH.HTML], page 17,

### 8.430.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /world/clock/calendar/now/fragment. It returns a content-type of text/html.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.430.3 File

Defined in file src/endpoints/slash-world.lisp.

## 8.431 Tootsville::Endpoint-Get-/ World/ Clock/ Calendar/ Year/ Year/ Fragment $\mapsto$ Html

### 8.431.1 Function

Endpoint-Get-/ World/ Clock/ Calendar/ Year/ Year/ Fragment $\mapsto$ Html names a function, with lambda list (YEAR):

Get a calendar fragment in HTML for 12 months of YEAR.

Contains HTML fragments for each of the 12 months.

See Section 3.4 [CHCEROGRYLLUM CAL-MONTH.HTML], page 17,

### 8.431.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /world/clock/calendar/year/:year/fragment. It returns a content-type of text/html.

YEAR is a parameter from the URI.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.431.3 File

Defined in file src/endpoints/slash-world.lisp.

## 8.432 Tootsville::Endpoint-Get-/ World/ Clock/ Calendar/ Year/ Year/ Month/ Month/ Fragment $\mapsto$ Html

### 8.432.1 Function

Endpoint-Get-/ World/ Clock/ Calendar/ Year/ Year/ Month/ Month/ Fragment $\mapsto$ Html names a function, with lambda list (YEAR MONTH):

Get a calendar fragment in HTML for MONTH of YEAR.

The HTML fragment contains the month header, day-of-week headers, and has holidays marked with HTML title attributes.

See Section 3.4 [CHCEROGRYLLUM CAL-MONTH.HTML], page 17,

### 8.432.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /world/clock/calendar/year/:year/month/:month/fragment. It returns a content-type of text/html.

The URI includes parameters: YEAR, MONTH.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.432.3 File

Defined in file src/endpoints/slash-world.lisp.

## 8.433 Tootsville::Endpoint-Get-/ World/ Clock/ Calendar/ Year/ Year/ Month/ Month→Html

### 8.433.1 Function

Endpoint-Get-/ World/ Clock/ Calendar/ Year/ Year/ Month/ Month→Html names a function, with lambda list (YEAR MONTH):

Get a calendar as an HTML page for MONTH of YEAR.

Produces a minimal HTML page framework surrounding a single-month HTML calendar.

See Section 3.4 [CHCEROGRYLLUM CAL-MONTH.HTML], page 17.

### 8.433.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /world/clock/calendar/year/:year/month/:month. It returns a content-type of text/html.

The URI includes parameters: YEAR, MONTH.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.433.3 File

Defined in file src/endpoints/slash-world.lisp.



## 8.434 Tootsville::Endpoint-Get-/ World/ Clock/ Date/ Abbrev $\mapsto$ Txt

### 8.434.1 Function

Endpoint-Get-/ World/ Clock/ Date/ Abbrev $\mapsto$ Txt names a function, with lambda list NIL:

Get the date on Chœrogrillum (abbreviated date string)

### 8.434.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /world/clock/date/abbrev. It returns a content-type of text/plain.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.434.3 File

Defined in file src/endpoints/slash-world.lisp.

## 8.435 Tootsville::Endpoint-Get-/ World/ Clock/ Date/ Long $\mapsto$ Txt

### 8.435.1 Function

Endpoint-Get-/ World/ Clock/ Date/ Long $\mapsto$ Txt names a function, with lambda list NIL:

Get the date on Chœrogrillum (pretty-printed date string)

### 8.435.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /world/clock/date/long. It returns a content-type of text/plain.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.435.3 File

Defined in file src/endpoints/slash-world.lisp.

## 8.436 Tootsville::Endpoint-Get-/ World/ Clock/ Date $\mapsto$ Txt

### 8.436.1 Function

Endpoint-Get-/ World/ Clock/ Date $\mapsto$ Txt names a function, with lambda list NIL:

Get the date on Chœrogyllum (pretty-printed date string)

### 8.436.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /world/clock/date. It returns a content-type of text/plain.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.436.3 File

Defined in file src/endpoints/slash-world.lisp.

## 8.437 Tootsville::Endpoint-Get-/ World/ Clock/ Time/ Detailed $\mapsto$ Txt

### 8.437.1 Function

Endpoint-Get-/ World/ Clock/ Time/ Detailed $\mapsto$ Txt names a function, with lambda list NIL:

Get a long string explaining the date, time, and other info.

See Section 8.369 [TOOTSVILLE DETAILED-TIME], page 625.

### 8.437.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /world/clock/time/detailed. It returns a content-type of text/plain.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.437.3 File

Defined in file src/endpoints/slash-world.lisp.

## 8.438 Tootsville::Endpoint-Get-/ World/ Clock/ Time→Json

### 8.438.1 Function

Endpoint-Get-/ World/ Clock/ Time→Json names a function, with lambda list NIL:

Get the date & time on Chœrogrillum as a JSON structure.

The returned object will have the following keys:

<code>sec</code>	Seconds into the minute
<code>min</code>	Minutes into the hour
<code>hour</code>	Hour of the day
<code>day</code>	Day of the month
<code>month</code>	Month of the year
<code>year</code>	Year
<code>weekday</code>	Day of the week (numeric)
<code>otherMonthDay</code>	Day of the Other Moon month
<code>pinkMonthDay</code>	Day of the Pink Moon month
<code>julian</code>	Julian day
<code>julian360</code>	Julian day of the year
<code>holiday</code>	Name of any holiday that occurs on this day

### 8.438.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /world/clock/time. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.438.3 File

Defined in file src/endpoints/slash-world.lisp.

## 8.439 Tootsville::Endpoint-Get-/ World/ Clock/ Time $\mapsto$ Txt

### 8.439.1 Function

Endpoint-Get-/ World/ Clock/ Time $\mapsto$ Txt names a function, with lambda list NIL:

Get the current time on Choerogryllum (time string with seconds)

Returns a string of Hours:Minutes:Seconds.

### 8.439.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /world/clock/time. It returns a content-type of text/plain.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.439.3 File

Defined in file src/endpoints/slash-world.lisp.

## 8.440 Tootsville::Endpoint-Get-/ World/ Sky/ Tootanga/ Latitude/ Longitude→Json

### 8.440.1 Function

Endpoint-Get-/ World/ Sky/ Tootanga/ Latitude/ Longitude→Json names a function, with lambda list (LATITUDE LONGITUDE):

Get the contents of the sky visible over (LATITUDE, LONGITUDE).

This data includes the position of the Sun (which could be below the horizon), the position of each moon, and the (fractional) phase of that moon. It may also include an array of clouds, precipitation (rain or snow), lightning patterns, &c.

This will *not* include things that are flying in the sky.

### 8.440.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /world/sky/tootanga/:latitude/:longitude. It returns a content-type of application/json.

The URI includes parameters: LATITUDE, LONGITUDE.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.440.3 File

Defined in file src/endpoints/slash-world.lisp.

## 8.441 Tootsville::Endpoint-Get-/ World/ Tootanga/ Latitude/ Longitude/ Altitude→Json

### 8.441.1 Function

Endpoint-Get-/ World/ Tootanga/ Latitude/ Longitude/ Altitude→Json names a function, with lambda list (LATITUDE LONGITUDE ALTITUDE):

Get the information about the area near (LATITUDE, LONGITUDE, ALTITUDE)

The terrain and objects in that area, characters, &c. will be returned.

Your character must be able to observe that general area. No peeking!

### 8.441.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /world/tootanga/:latitude/:longitude/:altitude. It returns a content-type of application/json.

The URI includes parameters: LATITUDE, LONGITUDE, ALTITUDE.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.441.3 File

Defined in file src/endpoints/slash-world.lisp.



## 8.442 Tootsville::Endpoint-Get-/ World $\mapsto$ Json

### 8.442.1 Function

Endpoint-Get-/ World $\mapsto$ Json names a function, with lambda list NIL:

Get world-related info in general. Not implemented.

### 8.442.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /world. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.442.3 File

Defined in file src/endpoints/slash-world.lisp.

## 8.443 Tootsville::Endpoint-Get-/ $\mapsto$ Html

### 8.443.1 Function

Endpoint-Get-/  $\mapsto$ Html names a function, with lambda list NIL:

GET on the root redirects to the main web page for the cluster (eg, <https://Tootsville.org/>)

### 8.443.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method GET at the URI template /. It returns a content-type of text/html.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.443.3 File

Defined in file src/web.lisp.

## 8.444 Tootsville::Endpoint-Hash

### 8.444.1 Function

Endpoint-Hash names an undocumented function, with lambda list (ENDPOINT-IDENTIFIER).

### 8.444.2 File

Defined in file src/endpoint.lisp.

## **8.445 Tootsville::Endpoint-Kinda-Key**

### **8.445.1 Function**

Endpoint-Kinda-Key names an undocumented function, with lambda list (ENDPOINT).

### **8.445.2 File**

Defined in file src/endpoint.lisp.

## **8.446 Tootsville::Endpoint-Method**

### **8.446.1 Function**

Endpoint-Method names an undocumented function, with lambda list (OBJECT).

## **8.447 Tootsville::Endpoint-Patch-/ Users/ Me→Json**

### **8.447.1 Function**

Endpoint-Patch-/ Users/ Me→Json names a function, with lambda list NIL:

Alters information about your user account.

Requires player authentication.

Requires a body with fields to be changed, and their new values. TODO.

### **8.447.2 Status: 200 OK**

### **8.447.3 Status: 401 Authorization Required**

### **8.447.4 Status: 403 Authorization Failed**

### **8.447.5 Status: 405 Not Allowed**

### **8.447.6 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method PATCH at the URI template /users/me. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.447.7 File**

Defined in file src/endpoints/slash-users.lisp.

## 8.448 Tootsville::Endpoint-Post-/ Gossip/ Alexa/ Chat/ Region/ Region $\mapsto$ Json

### 8.448.1 Function

Endpoint-Post-/ Gossip/ Alexa/ Chat/ Region/ Region $\mapsto$ Json names a function, with lambda list (REGION):

Undocumented endpoint for POST /gossip/alexa/chat/region/:region  $\mapsto$  :APPLICATION/JSON

### 8.448.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /gossip/alexa/chat/region/:region. It returns a content-type of application/json.

REGION is a parameter from the URI.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.448.3 File

Defined in file src/endpoints/gossip/alexa/chat.lisp.

## 8.449 Tootsville::Endpoint-Post-/ Gossip/ Alexa/ Clock/ Region/ Region $\mapsto$ Json

### 8.449.1 Function

Endpoint-Post-/ Gossip/ Alexa/ Clock/ Region/ Region $\mapsto$ Json names a function, with lambda list (REGION):

Undocumented endpoint for POST /gossip/alex/clock/region/:region  $\mapsto$  :APPLICATION/JSON

### 8.449.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /gossip/alex/clock/region/:region. It returns a content-type of application/json.

REGION is a parameter from the URI.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.449.3 File

Defined in file src/endpoints/gossip/alex/clock.lisp.



## 8.450 Tootsville::Endpoint-Post-/ Gossip/ Alexa/ Info/ Region/ Region $\mapsto$ Json

### 8.450.1 Function

Endpoint-Post-/ Gossip/ Alexa/ Info/ Region/ Region $\mapsto$ Json names a function, with lambda list (REGION):

Undocumented endpoint for POST /gossip/alex/info/region/:region  $\mapsto$  :APPLICATION/JSON

### 8.450.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /gossip/alex/info/region/:region. It returns a content-type of application/json.

REGION is a parameter from the URI.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.450.3 File

Defined in file src/endpoints/gossip/alex/info.lisp.

## 8.451 Tootsville::Endpoint-Post-/ Gossip/ Answers/ Uuid $\mapsto$ Sdp

### 8.451.1 Function

Endpoint-Post-/ Gossip/ Answers/ Uuid $\mapsto$ Sdp names a function, with lambda list (UUID):

Post an answer to a received SDP block.

The client, having received an SDP offer, computes an SDP answer and posts it back to this endpoint.

### 8.451.2 202 Accepted

The posted data has been accepted and will be relayed back to the offeror.

### 8.451.3 404 Not Found

The UUID given is not associated with an outstanding offer.

### 8.451.4 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /gossip/answers/:uuid. It returns a content-type of application/sdp.

UUID is a parameter from the URI.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.451.5 File

Defined in file src/endpoints/slash-gossip.lisp.

## 8.452 Tootsville::Endpoint-Post-/ Gossip/ Offers $\mapsto$ Sdp

### 8.452.1 Function

Endpoint-Post-/ Gossip/ Offers $\mapsto$ Sdp names a function, with lambda list NIL:

Provide a new offer. Body is an SDP offer. Reply will be an offer URI.

The offer URI will be needed to retrieve the answer to your offer from whatever peer may accept it. There is no guarantee that an offer will be accepted.

### 8.452.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /gossip/offers. It returns a content-type of application/sdp.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.452.3 File

Defined in file src/endpoints/slash-gossip.lisp.

## 8.453 Tootsville::Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Call→Xml

### 8.453.1 Function

Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Call→Xml names a function, with lambda list NIL:

Respond to a phone call to NUMBER at Twilio.

Someone has called us at NUMBER, and Twilio needs to know how to reply. Send an XML (Twiml) response.

### 8.453.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /gossip/twilio/incoming/call. It returns a content-type of text/xml.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.453.3 File

Defined in file src/endpoints/gossip/twilio.lisp.

## 8.454 Tootsville::Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Fax→Xml

### 8.454.1 Function

Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Fax→Xml names a function, with lambda list NIL:

Respond to a fax call to NUMBER at Twilio.

Someone has faxxed us at NUMBER, and Twilio needs to know how to reply. Send an XML (Twiml) response.

### 8.454.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /gossip/twilio/incoming/fax. It returns a content-type of text/xml.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.454.3 File

Defined in file src/endpoints/gossip/twilio.lisp.

## 8.455 Tootsville::Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Sms→Xml

### 8.455.1 Function

Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Sms→Xml names a function, with lambda list NIL:

Respond to an SMS or MMS message to NUMBER at Twilio.

Someone has messaged us at NUMBER, and Twilio needs to know how to reply. Send an XML (TwiML) response.

### 8.455.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /gossip/twilio/incoming/sms. It returns a content-type of text/xml.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.455.3 File

Defined in file src/endpoints/gossip/twilio.lisp.

## 8.456 Tootsville::Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Verify $\mapsto$ Xml

### 8.456.1 Function

Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Verify $\mapsto$ Xml names a function, with lambda list NIL:

Check a Verify code from a user's phone.

We have sent a Verify code to someone through Twilio. They have replied by entering that code, which we now need to verify through the Authy Verify endpoint.

### 8.456.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /gossip/twilio/incoming/verify. It returns a content-type of text/xml.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.456.3 File

Defined in file src/endpoints/gossip/twilio.lisp.

## 8.457 Tootsville::Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Whatsapp→Xml

### 8.457.1 Function

Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Whatsapp→Xml names a function, with lambda list NIL:

Respond to a WhatsApp message to NUMBER at Twilio.

Someone has messaged us at NUMBER, and Twilio needs to know how to reply. Send an XML (TwiML) response.

### 8.457.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /gossip/twilio/incoming/whatsapp. It returns a content-type of text/xml.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.457.3 File

Defined in file src/endpoints/gossip/twilio.lisp.



## 8.458 Tootsville::Endpoint-Post-/ Login/ Child→Json

### 8.458.1 Function

Endpoint-Post-/ Login/ Child→Json names a function, with lambda list NIL:

Child login submission.

See Section 8.828 [TOOTSVILLE LOGIN-CHILD], page 1113, for details of the child login protocol.

### 8.458.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /login/child. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.458.3 File

Defined in file src/endpoints/slash-login.lisp.

## **8.459 Tootsville::Endpoint-Post-/ Maintenance/ Buildapp/ Status $\mapsto$ Nil**

### **8.459.1 Function**

Endpoint-Post-/ Maintenance/ Buildapp/ Status $\mapsto$ Nil names a function, with lambda list NIL:

Checking on the last BuildApp request

### **8.459.2 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /maintenance/buildapp/status. It returns a content-type of nil.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.459.3 File**

Defined in file src/endpoints/slash-maintenance.lisp.

## 8.460 Tootsville::Endpoint-Post-/ Maintenance/ Buildapp $\mapsto$ Nil

### 8.460.1 Function

Endpoint-Post-/ Maintenance/ Buildapp $\mapsto$ Nil names a function, with lambda list NIL:

Recompiling Tootsville executable

### 8.460.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /maintenance/buildapp. It returns a content-type of nil.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.460.3 File

Defined in file src/endpoints/slash-maintenance.lisp.

## **8.461 Tootsville::Endpoint-Post-/ Maintenance/ Hot-Reload $\mapsto$ Nil**

### **8.461.1 Function**

Endpoint-Post-/ Maintenance/ Hot-Reload $\mapsto$ Nil names a function, with lambda list NIL:  
Reloading from local sources

### **8.461.2 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /maintenance/hot-reload. It returns a content-type of nil.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.461.3 File**

Defined in file src/endpoints/slash-maintenance.lisp.

## 8.462 Tootsville::Endpoint-Post-/ Maintenance/ Quicklisp-Update $\mapsto$ Nil

### 8.462.1 Function

Endpoint-Post-/ Maintenance/ Quicklisp-Update $\mapsto$ Nil names a function, with lambda list NIL:

Updating the Quicklisp client and distributions

### 8.462.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /maintenance/quicklisp-update. It returns a content-type of nil.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.462.3 File

Defined in file src/endpoints/slash-maintenance.lisp.

## 8.463 Tootsville::Endpoint-Post-/ Maintenance/ Quit $\mapsto$ Nil

### 8.463.1 Function

Endpoint-Post-/ Maintenance/ Quit $\mapsto$ Nil names a function, with lambda list NIL:

Quit running

### 8.463.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /maintenance/quit. It returns a content-type of nil.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.463.3 File

Defined in file src/endpoints/slash-maintenance.lisp.

## 8.464 Tootsville::Endpoint-Post-/ Maintenance/ Reload-Jscl→Nil

### 8.464.1 Function

Endpoint-Post-/ Maintenance/ Reload-Jscl→Nil names a function, with lambda list NIL:  
Recompiling jscl.js

### 8.464.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /maintenance/reload-jscl. It returns a content-type of nil.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.464.3 File

Defined in file src/endpoints/slash-maintenance.lisp.

## 8.465 Tootsville::Endpoint-Post-/ Toots $\mapsto$ Json

### 8.465.1 Function

Endpoint-Post-/ Toots $\mapsto$ Json names a function, with lambda list NIL:

Create a new Toot.

Input JSON must have the following fields: name, baseColor, padColor, pattern, patternColor, tShirtColor

Responds with 201 (Created); or 409 (Conflict) if the name is in use or for some other reason the value can't be entered; 422 if the Toot name, color or pattern name(s) given are not valid. (400 if the request is malformed.)

### 8.465.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /toots. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.465.3 File

Defined in file src/endpoints/slash-toots.lisp.



## 8.466 Tootsville::Endpoint-Post-/ Users/ Me/ Play-With/ Toot-Name→Json

### 8.466.1 Function

Endpoint-Post-/ Users/ Me/ Play-With/ Toot-Name→Json names a function, with lambda list (TOOT-NAME):

Begin playing with the Toot named TOOT-NAME.

`'Toot-Name'`

The name of the Toot character to play with.

### 8.466.2 Status: 200 OK

You are now in control of this Toot. The Toot's info will be returned.

The returned body will be a JSON object with two keys;

`toot` The Toot avatar information as returned by Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534.

`player` The player information returned by Section 8.965 [TOOTSVILLE PERSON-INFO], page 1250.

### 8.466.3 Status: 401 Authorization Required

No user credentials were passed.

### 8.466.4 Status: 403 Authorization Failed

The user credentials presented were not recognized.

### 8.466.5 Status: 404 Not Found

The Toot named does not exist.

### 8.466.6 Status: 405 Not Allowed

The Toot named is one that you have permission to use, but are not the main owner of. This is usually a child account.

### 8.466.7 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template `/users/me/play-with/:toot-name`. It returns a content-type of `application/json`.

TOOT-NAME is a parameter from the URI.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.466.8 File

Defined in file `src/endpoints/slash-users.lisp`.

## 8.467 Tootsville::Endpoint-Post-/ World/ Infinity/ Add-Furniture→Json

### 8.467.1 Function

Endpoint-Post-/ World/ Infinity/ Add-Furniture→Json names a function, with lambda list NIL:

Alias for INFINITY-SET-FURNITURE.

### 8.467.2 Infinity Mode command

See Section 8.678 [TOOTSVILLE INFINITY-ADD-FURNITURE], page 936,

### 8.467.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/add-furniture. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.467.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.468 Tootsville::Endpoint-Post-/ World/ Infinity/ Add-Journal-Entry $\mapsto$ Json

### 8.468.1 Function

Endpoint-Post-/ World/ Infinity/ Add-Journal-Entry $\mapsto$ Json names a function, with lambda list NIL:

Add a staff journal entry.

### 8.468.2 Infinity Mode command

See Section 8.679 [TOOTSVILLE INFINITY-ADD-JOURNAL-ENTRY], page 937,

### 8.468.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/add-journal-entry. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.468.4 File

Defined in file src/infinity/tootsville-commands.lisp.

## 8.469 Tootsville::Endpoint-Post-/ World/ Infinity/ Add-To-List $\mapsto$ Json

### 8.469.1 Function

Endpoint-Post-/ World/ Infinity/ Add-To-List $\mapsto$ Json names a function, with lambda list NIL:

Add a user to a buddy list or ignore list (removed in 1.2)

### 8.469.2 Infinity Mode command

See Section 8.680 [TOOTSVILLE INFINITY-ADD-TO-LIST], page 938,

### 8.469.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/add-to-list. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.469.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.470 Tootsville::Endpoint-Post-/ World/ Infinity/ Click→Json

### 8.470.1 Function

Endpoint-Post-/ World/ Infinity/ Click→Json names a function, with lambda list NIL:

Used by the client to report a mouse click or finger tap.

### 8.470.2 Infinity Mode command

See Section 8.681 [TOOTSVILLE INFINITY-CLICK], page 939,

### 8.470.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/click. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.470.4 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.471 Tootsville::Endpoint-Post-/ World/ Infinity/ Consider-Child-Approval→Json**

### **8.471.1 Function**

Endpoint-Post-/ World/ Infinity/ Consider-Child-Approval→Json names a function, with lambda list NIL:

Consider whether to approve a child's request with ID UUID.

### **8.471.2 Infinity Mode command**

See Section 8.682 [TOOTSVILLE INFINITY-CONSIDER-CHILD-APPROVAL], page 941,

### **8.471.3 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/consider-child-approval. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.471.4 File**

Defined in file src/infinity/new-commands-20.lisp.

## 8.472 Tootsville::Endpoint-Post-/ World/ Infinity/ Create-User-House $\mapsto$ Json

### 8.472.1 Function

Endpoint-Post-/ World/ Infinity/ Create-User-House $\mapsto$ Json names a function, with lambda list NIL:

Either claim the user's house and lot, or add a room to their house.

### 8.472.2 Infinity Mode command

See Section 8.683 [TOOTSVILLE INFINITY-CREATE-USER-HOUSE], page 942,

### 8.472.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/create-user-house. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.472.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.473 Tootsville::Endpoint-Post-/ World/ Infinity/ Delete-Mail-Message→Json

### 8.473.1 Function

Endpoint-Post-/ World/ Infinity/ Delete-Mail-Message→Json names a function, with lambda list NIL:

Delete a message from the user's (SMS) mailbox

### 8.473.2 Infinity Mode command

See Section 8.684 [TOOTSVILLE INFINITY-DELETE-MAIL-MESSAGE], page 943,

### 8.473.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/delete-mail-message. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.473.4 File

Defined in file src/infinity/tootsville-commands.lisp.



## 8.474 Tootsville::Endpoint-Post-/ World/ Infinity/ Dofft→Json

### 8.474.1 Function

Endpoint-Post-/ World/ Infinity/ Dofft→Json names a function, with lambda list NIL:

Doff all clothing items.

### 8.474.2 Infinity Mode command

See Section 8.686 [TOOTSVILLE INFINITY-DOFFF], page 945,

### 8.474.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/doff. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.474.4 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.475 Tootsville::Endpoint-Post-/ World/ Infinity/ Doff→Json**

### **8.475.1 Function**

Endpoint-Post-/ World/ Infinity/ Doff→Json names a function, with lambda list NIL:  
Remove clothes or Pivitz.

### **8.475.2 Infinity Mode command**

See Section 8.685 [TOOTSVILLE INFINITY-DOFF], page 944,

### **8.475.3 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/doff. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.475.4 File**

Defined in file src/infinity/tootsville-commands.lisp.

## 8.476 Tootsville::Endpoint-Post-/ World/ Infinity/ Don $\mapsto$ Json

### 8.476.1 Function

Endpoint-Post-/ World/ Infinity/ Don $\mapsto$ Json names a function, with lambda list NIL:  
Don (or equip) an item

### 8.476.2 Infinity Mode command

See Section 8.687 [TOOTSVILLE INFINITY-DON], page 946,

### 8.476.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/don. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.476.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.477 Tootsville::Endpoint-Post-/ World/ Infinity/ Echo $\mapsto$ Json

### 8.477.1 Function

Endpoint-Post-/ World/ Infinity/ Echo $\mapsto$ Json names a function, with lambda list NIL:

Echoes back the supplied JSON (or ActionScript) object to the client.

### 8.477.2 Infinity Mode command

See Section 8.688 [TOOTSVILLE INFINITY-ECHO], page 948,

### 8.477.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/echo. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.477.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.478 Tootsville::Endpoint-Post-/ World/ Infinity/ End-Event $\mapsto$ Json

### 8.478.1 Function

Endpoint-Post-/ World/ Infinity/ End-Event $\mapsto$ Json names a function, with lambda list NIL:

Attempt to end an event.

End an event begun by Section 8.738 [TOOTSVILLE INFINITY-START-EVENT], page 1020, q.v.

### 8.478.2 Infinity Mode command

See Section 8.689 [TOOTSVILLE INFINITY-END-EVENT], page 949,

### 8.478.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/end-event. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.478.4 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.479 Tootsville::Endpoint-Post-/ World/ Infinity/ Enumerate-Wear-Slots $\mapsto$ Json**

### **8.479.1 Function**

Endpoint-Post-/ World/ Infinity/ Enumerate-Wear-Slots $\mapsto$ Json names a function, with lambda list NIL:

Enumerates all possible wear slots for any avatar.

### **8.479.2 Infinity Mode command**

See Section 8.690 [TOOTSVILLE INFINITY-ENUMERATE-WEAR-SLOTS], page 952,

### **8.479.3 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/enumerate-wear-slots. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.479.4 File**

Defined in file src/infinity/new-commands-20.lisp.

## 8.480 Tootsville::Endpoint-Post-/ World/ Infinity/ Finger $\mapsto$ Json

### 8.480.1 Function

Endpoint-Post-/ World/ Infinity/ Finger $\mapsto$ Json names a function, with lambda list NIL:

Get public info for a list of Toots.

### 8.480.2 Infinity Mode command

See Section 8.691 [TOOTSVILLE INFINITY-FINGER], page 953,

### 8.480.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/finger. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.480.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.481 Tootsville::Endpoint-Post-/ World/ Infinity/ Game-Action $\mapsto$ Json

### 8.481.1 Function

Endpoint-Post-/ World/ Infinity/ Game-Action $\mapsto$ Json names a function, with lambda list NIL:

Send an in-world game's action.

### 8.481.2 Infinity Mode command

See Section 8.692 [TOOTSVILLE INFINITY-GAME-ACTION], page 954,

### 8.481.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/game-action. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.481.4 File

Defined in file src/infinity/legacy-commands.lisp.



## 8.482 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Avatars→Json

### 8.482.1 Function

Endpoint-Post-/ World/ Infinity/ Get-Avatars→Json names a function, with lambda list NIL:

Get avatar data for a list of (other) users.

### 8.482.2 Infinity Mode command

See Section 8.694 [TOOTSVILLE INFINITY-GET-AVATARS], page 959,

### 8.482.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-avatars. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.482.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.483 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Color-Palettes $\mapsto$ Json

### 8.483.1 Function

Endpoint-Post-/ World/ Infinity/ Get-Color-Palettes $\mapsto$ Json names a function, with lambda list NIL:

```
getColorPalettes
```

### 8.483.2 Infinity Mode command

See Section 8.695 [TOOTSVILLE INFINITY-GET-COLOR-PALETTES], page 960,

### 8.483.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-color-palettes. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.483.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.484 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Inventory-By-Type $\mapsto$ Json

### 8.484.1 Function

Endpoint-Post-/ World/ Infinity/ Get-Inventory-By-Type $\mapsto$ Json names a function, with lambda list NIL:

Get a subset of items from your own inventory

### 8.484.2 Infinity Mode command

See Section 8.697 [TOOTSVILLE INFINITY-GET-INVENTORY-BY-TYPE], page 962,

### 8.484.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-inventory-by-type. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.484.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.485 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Inventory $\mapsto$ Json

### 8.485.1 Function

Endpoint-Post-/ World/ Infinity/ Get-Inventory $\mapsto$ Json names a function, with lambda list NIL:

Get all inventory for an user (themselves) — both active and inactive

### 8.485.2 Infinity Mode command

See Section 8.696 [TOOTSVILLE INFINITY-GET-INVENTORY], page 961,

### 8.485.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-inventory. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.485.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.486 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Mail-In-Box $\mapsto$ Json

### 8.486.1 Function

Endpoint-Post-/ World/ Infinity/ Get-Mail-In-Box $\mapsto$ Json names a function, with lambda list NIL:

Get a listing of messages in an SMS mailbox.

### 8.486.2 Infinity Mode command

See Section 8.698 [TOOTSVILLE INFINITY-GET-MAIL-IN-BOX], page 964,

### 8.486.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-mail-in-box. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.486.4 File

Defined in file src/infinity/tootsville-commands.lisp.

## 8.487 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Online-Users→Json

### 8.487.1 Function

Endpoint-Post-/ World/ Infinity/ Get-Online-Users→Json names a function, with lambda list NIL:

Get a list of users online.

### 8.487.2 Infinity Mode command

See Section 8.699 [TOOTSVILLE INFINITY-GET-ONLINE-USERS], page 966,

### 8.487.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-online-users. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.487.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.488 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Passport $\mapsto$ Json

### 8.488.1 Function

Endpoint-Post-/ World/ Infinity/ Get-Passport $\mapsto$ Json names a function, with lambda list NIL:

Get the list of places that the user has gotten a passport stamp at.

### 8.488.2 Infinity Mode command

See Section 8.700 [TOOTSVILLE INFINITY-GET-PASSPORT], page 967,

### 8.488.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-passport. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.488.4 File

Defined in file src/infinity/tootsville-commands.lisp.

## 8.489 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Room-List $\mapsto$ Json

### 8.489.1 Function

Endpoint-Post-/ World/ Infinity/ Get-Room-List $\mapsto$ Json names a function, with lambda list NIL:

Get a list of all “well known” Rooms currently active/visible.

### 8.489.2 Infinity Mode command

See Section 8.701 [TOOTSVILLE INFINITY-GET-ROOM-LIST], page 968,

### 8.489.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-room-list. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.489.4 File

Defined in file src/infinity/legacy-commands.lisp.



## 8.490 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Room-Vars $\mapsto$ Json

### 8.490.1 Function

Endpoint-Post-/ World/ Infinity/ Get-Room-Vars $\mapsto$ Json names a function, with lambda list NIL:

Returns “room variables.”

### 8.490.2 Infinity Mode command

See Section 8.702 [TOOTSVILLE INFINITY-GET-ROOM-VARS], page 969,

### 8.490.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-room-vars. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.490.4 File

Defined in file src/infinity/new-commands-20.lisp.

## 8.491 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Server-Time $\mapsto$ Json

### 8.491.1 Function

Endpoint-Post-/ World/ Infinity/ Get-Server-Time $\mapsto$ Json names a function, with lambda list NIL:

Send the server time to the client requesting it

### 8.491.2 Infinity Mode command

See Section 8.703 [TOOTSVILLE INFINITY-GET-SERVER-TIME], page 973,

### 8.491.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-server-time. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.491.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.492 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Session-Apple→Json

### 8.492.1 Function

Endpoint-Post-/ World/ Infinity/ Get-Session-Apple→Json names a function, with lambda list NIL:

Initialise a session key for stream or batch mode operations.

### 8.492.2 Infinity Mode command

See Section 8.704 [TOOTSVILLE INFINITY-GET-SESSION-APPLE], page 974,

### 8.492.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-session-apple. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.492.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.493 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Store-Item-Info→Json

### 8.493.1 Function

Endpoint-Post-/ World/ Infinity/ Get-Store-Item-Info→Json names a function, with lambda list NIL:

Get information about items in a store which can be purchased.

### 8.493.2 Infinity Mode command

See Section 8.705 [TOOTSVILLE INFINITY-GET-STORE-ITEM-INFO], page 975,

### 8.493.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-store-item-info. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.493.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.494 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-User-Lists $\mapsto$ Json

### 8.494.1 Function

Endpoint-Post-/ World/ Infinity/ Get-User-Lists $\mapsto$ Json names a function, with lambda list NIL:

Get the user's buddy list and ignore list.

### 8.494.2 Infinity Mode command

See Section 8.706 [TOOTSVILLE INFINITY-GET-USER-LISTS], page 976,

### 8.494.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-user-lists. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.494.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.495 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Wallet $\mapsto$ Json

### 8.495.1 Function

Endpoint-Post-/ World/ Infinity/ Get-Wallet $\mapsto$ Json names a function, with lambda list NIL:

Get the contents of the player's wallet (peanuts and fairy dust)

### 8.495.2 Infinity Mode command

See Section 8.707 [TOOTSVILLE INFINITY-GET-WALLET], page 977,

### 8.495.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-wallet. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.495.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.496 Tootsville::Endpoint-Post-/ World/ Infinity/ Get-Zone-List $\mapsto$ Json

### 8.496.1 Function

Endpoint-Post-/ World/ Infinity/ Get-Zone-List $\mapsto$ Json names a function, with lambda list NIL:

Get a list of all Zones currently active/visible.

### 8.496.2 Infinity Mode command

See Section 8.708 [TOOTSVILLE INFINITY-GET-ZONE-LIST], page 978,

### 8.496.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/get-zone-list. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.496.4 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.497 Tootsville::Endpoint-Post-/ World/ Infinity/ Give $\mapsto$ Json**

### **8.497.1 Function**

Endpoint-Post-/ World/ Infinity/ Give $\mapsto$ Json names a function, with lambda list NIL:

Give an item to another user.

### **8.497.2 Infinity Mode command**

See Section 8.709 [TOOTSVILLE INFINITY-GIVE], page 979,

### **8.497.3 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/give. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.497.4 File**

Defined in file src/infinity/legacy-commands.lisp.



## 8.498 Tootsville::Endpoint-Post-/ World/ Infinity/ Go→Json

### 8.498.1 Function

Endpoint-Post-/ World/ Infinity/ Go→Json names a function, with lambda list NIL:

go to a place and/or perform a gesture

### 8.498.2 Infinity Mode command

See Section 8.710 [TOOTSVILLE INFINITY-GO], page 980,

### 8.498.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/go. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.498.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.499 Tootsville::Endpoint-Post-/ World/ Infinity/ Init-User-Room $\mapsto$ Json

### 8.499.1 Function

Endpoint-Post-/ World/ Infinity/ Init-User-Room $\mapsto$ Json names a function, with lambda list NIL:

Create a user's private room (in their house).

### 8.499.2 Infinity Mode command

See Section 8.711 [TOOTSVILLE INFINITY-INIT-USER-ROOM], page 981,

### 8.499.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/init-user-room. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.499.4 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.500 Tootsville::Endpoint-Post-/ World/ Infinity/ Join→Json**

### **8.500.1 Function**

Endpoint-Post-/ World/ Infinity/ Join→Json names a function, with lambda list NIL:  
Join a room.

### **8.500.2 Infinity Mode command**

See Section 8.712 [TOOTSVILLE INFINITY-JOIN], page 982,

### **8.500.3 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/join. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.500.4 File**

Defined in file src/infinity/legacy-commands.lisp.

## **8.501 Tootsville::Endpoint-Post-/ World/ Infinity/ Logout $\mapsto$ Json**

### **8.501.1 Function**

Endpoint-Post-/ World/ Infinity/ Logout $\mapsto$ Json names a function, with lambda list NIL:  
Log out of this game session

### **8.501.2 Infinity Mode command**

See Section 8.714 [TOOTSVILLE INFINITY-LOGOUT], page 985,

### **8.501.3 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/logout. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.501.4 File**

Defined in file src/infinity/legacy-commands.lisp.

## 8.502 Tootsville::Endpoint-Post-/ World/ Infinity/ Mail-Customer-Service $\mapsto$ Json

### 8.502.1 Function

Endpoint-Post-/ World/ Infinity/ Mail-Customer-Service $\mapsto$ Json names a function, with lambda list NIL:

Send an eMail to customer service (feedback)

### 8.502.2 Infinity Mode command

See Section 8.715 [TOOTSVILLE INFINITY-MAIL-CUSTOMER-SERVICE], page 986,

### 8.502.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/mail-customer-service. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.502.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.503 Tootsville::Endpoint-Post-/ World/ Infinity/ Peek-At-Inventory $\mapsto$ Json

### 8.503.1 Function

Endpoint-Post-/ World/ Infinity/ Peek-At-Inventory $\mapsto$ Json names a function, with lambda list NIL:

Look at other users' inventories

### 8.503.2 Infinity Mode command

See Section 8.716 [TOOTSVILLE INFINITY-PEEK-AT-INVENTORY], page 987,

### 8.503.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/peek-at-inventory. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.503.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.504 Tootsville::Endpoint-Post-/ World/ Infinity/ Ping $\mapsto$ Json

### 8.504.1 Function

Endpoint-Post-/ World/ Infinity/ Ping $\mapsto$ Json names a function, with lambda list NIL:

Send a ping to the server to get back a pong.

### 8.504.2 Infinity Mode command

See Section 8.717 [TOOTSVILLE INFINITY-PING], page 988,

### 8.504.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/ping. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.504.4 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.505 Tootsville::Endpoint-Post-/ World/ Infinity/ Play-With→Json**

### **8.505.1 Function**

Endpoint-Post-/ World/ Infinity/ Play-With→Json names a function, with lambda list NIL:

Choose a Toot as your active CHARACTER in the game.

### **8.505.2 Infinity Mode command**

See Section 8.718 [TOOTSVILLE INFINITY-PLAY-WITH], page 989,

### **8.505.3 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/play-with. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.505.4 File**

Defined in file src/infinity/new-commands-20.lisp.



## 8.506 Tootsville::Endpoint-Post-/ World/ Infinity/ Prompt-Reply $\mapsto$ Json

### 8.506.1 Function

Endpoint-Post-/ World/ Infinity/ Prompt-Reply $\mapsto$ Json names a function, with lambda list NIL:

Accept a reply to a server-initiated prompt

### 8.506.2 Infinity Mode command

See Section 8.720 [TOOTSVILLE INFINITY-PROMPT-REPLY], page 991,

### 8.506.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/prompt-reply. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.506.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.507 Tootsville::Endpoint-Post-/ World/ Infinity/ Quiesce $\mapsto$ Json

### 8.507.1 Function

Endpoint-Post-/ World/ Infinity/ Quiesce $\mapsto$ Json names a function, with lambda list NIL:

Quiesce Toot values to database for logout, or periodically as a backup.

### 8.507.2 Infinity Mode command

See Section 8.721 [TOOTSVILLE INFINITY-QUIESCE], page 994,

### 8.507.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/quiesce. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.507.4 File

Defined in file src/infinity/new-commands-20.lisp.

## 8.508 Tootsville::Endpoint-Post-/ World/ Infinity/ Read-Map $\mapsto$ Json

### 8.508.1 Function

Endpoint-Post-/ World/ Infinity/ Read-Map $\mapsto$ Json names a function, with lambda list NIL:

Get the positions of badges and named locations on the map.

### 8.508.2 Infinity Mode command

See Section 8.722 [TOOTSVILLE INFINITY-READ-MAP], page 995,

### 8.508.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/read-map. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.508.4 File

Defined in file src/infinity/new-commands-20.lisp.

## 8.509 Tootsville::Endpoint-Post-/ World/ Infinity/ Remove-From-List $\mapsto$ Json

### 8.509.1 Function

Endpoint-Post-/ World/ Infinity/ Remove-From-List $\mapsto$ Json names a function, with lambda list NIL:

Remove someone from a buddy list or ignore list.

### 8.509.2 Infinity Mode command

See Section 8.723 [TOOTSVILLE INFINITY-REMOVE-FROM-LIST], page 996,

### 8.509.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/remove-from-list. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.509.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.510 Tootsville::Endpoint-Post-/ World/ Infinity/ Report-Bug $\mapsto$ Json

### 8.510.1 Function

Endpoint-Post-/ World/ Infinity/ Report-Bug $\mapsto$ Json names a function, with lambda list NIL:

This method allows the client to “phone home” to report a bug.

### 8.510.2 Infinity Mode command

See Section 8.724 [TOOTSVILLE INFINITY-REPORT-BUG], page 997,

### 8.510.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/report-bug. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.510.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.511 Tootsville::Endpoint-Post-/ World/ Infinity/ Report-User $\mapsto$ Json

### 8.511.1 Function

Endpoint-Post-/ World/ Infinity/ Report-User $\mapsto$ Json names a function, with lambda list NIL:

Report an user to the moderator(s) on duty for breaking a rule

### 8.511.2 Infinity Mode command

See Section 8.725 [TOOTSVILLE INFINITY-REPORT-USER], page 1002,

### 8.511.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/report-user. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.511.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.512 Tootsville::Endpoint-Post-/ World/ Infinity/ Request-Buddy $\mapsto$ Json

### 8.512.1 Function

Endpoint-Post-/ World/ Infinity/ Request-Buddy $\mapsto$ Json names a function, with lambda list NIL:

Request adding a user to your buddy list (mutual-add) using the notification-based system

### 8.512.2 Infinity Mode command

See Section 8.726 [TOOTSVILLE INFINITY-REQUEST-BUDDY], page 1003,

### 8.512.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/request-buddy. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.512.4 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.513 Tootsville::Endpoint-Post-/ World/ Infinity/ Send-Mail-Message→Json**

### **8.513.1 Function**

Endpoint-Post-/ World/ Infinity/ Send-Mail-Message→Json names a function, with lambda list NIL:

Send an in-game SMS message.

### **8.513.2 Infinity Mode command**

See Section 8.727 [TOOTSVILLE INFINITY-SEND-MAIL-MESSAGE], page 1004,

### **8.513.3 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/send-mail-message. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.513.4 File**

Defined in file src/infinity/tootsville-commands.lisp.



## 8.514 Tootsville::Endpoint-Post-/ World/ Infinity/ Send-Out-Of-Band-Message→Json

### 8.514.1 Function

Endpoint-Post-/ World/ Infinity/ Send-Out-Of-Band-Message→Json names a function, with lambda list NIL:

Send an arbitrary JSON packet to another user, or all of the users

### 8.514.2 Infinity Mode command

See Section 8.728 [TOOTSVILLE INFINITY-SEND-OUT-OF-BAND-MESSAGE], page 1006,

### 8.514.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/send-out-of-band-message. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.514.4 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.515 Tootsville::Endpoint-Post-/ World/ Infinity/ Server-Time→Json**

### **8.515.1 Function**

Endpoint-Post-/ World/ Infinity/ Server-Time→Json names a function, with lambda list NIL:

Accept the client's notification of a server-time adjustment.

### **8.515.2 Infinity Mode command**

See Section 8.729 [TOOTSVILLE INFINITY-SERVER-TIME], page 1007,

### **8.515.3 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/server-time. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.515.4 File**

Defined in file src/infinity/legacy-commands.lisp.

## 8.516 Tootsville::Endpoint-Post-/ World/ Infinity/ Set-Avatar-Color $\mapsto$ Json

### 8.516.1 Function

Endpoint-Post-/ World/ Infinity/ Set-Avatar-Color $\mapsto$ Json names a function, with lambda list NIL:

Set the avatar base and extra (pad) colours for the given user.

### 8.516.2 Infinity Mode command

See Section 8.730 [TOOTSVILLE INFINITY-SET-AVATAR-COLOR], page 1008,

### 8.516.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/set-avatar-color. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.516.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.517 Tootsville::Endpoint-Post-/ World/ Infinity/ Set-Furniture→Json

### 8.517.1 Function

Endpoint-Post-/ World/ Infinity/ Set-Furniture→Json names a function, with lambda list NIL:

Set or change a furniture item.

### 8.517.2 Infinity Mode command

See Section 8.731 [TOOTSVILLE INFINITY-SET-FURNITURE], page 1009,

### 8.517.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/set-furniture. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.517.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.518 Tootsville::Endpoint-Post-/ World/ Infinity/ Set-Room-Var $\mapsto$ Json

### 8.518.1 Function

Endpoint-Post-/ World/ Infinity/ Set-Room-Var $\mapsto$ Json names a function, with lambda list NIL:

Set a room variable or set of room variables.

### 8.518.2 Infinity Mode command

See Section 8.732 [TOOTSVILLE INFINITY-SET-ROOM-VAR], page 1011,

### 8.518.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/set-room-var. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.518.4 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.519 Tootsville::Endpoint-Post-/ World/ Infinity/ Set-User-Var $\mapsto$ Json**

### **8.519.1 Function**

Endpoint-Post-/ World/ Infinity/ Set-User-Var $\mapsto$ Json names a function, with lambda list NIL:

Set “User Variables”

### **8.519.2 Infinity Mode command**

See Section 8.733 [TOOTSVILLE INFINITY-SET-USER-VAR], page 1012,

### **8.519.3 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/set-user-var. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.519.4 File**

Defined in file src/infinity/legacy-commands.lisp.

## 8.520 Tootsville::Endpoint-Post-/ World/ Infinity/ Shoot $\mapsto$ Json

### 8.520.1 Function

Endpoint-Post-/ World/ Infinity/ Shoot $\mapsto$ Json names a function, with lambda list NIL:

Fire a shot from a projectile device.

### 8.520.2 Infinity Mode command

See Section 8.734 [TOOTSVILLE INFINITY-SHOOT], page 1014,

### 8.520.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/shoot. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.520.4 File

Defined in file src/infinity/new-commands-20.lisp.

## **8.521 Tootsville::Endpoint-Post-/ World/ Infinity/ Spawn-Zone $\mapsto$ Json**

### **8.521.1 Function**

Endpoint-Post-/ World/ Infinity/ Spawn-Zone $\mapsto$ Json names a function, with lambda list NIL:

Spawn an additional server peer pairing.

### **8.521.2 Infinity Mode command**

See Section 8.735 [TOOTSVILLE INFINITY-SPAWN-ZONE], page 1015,

### **8.521.3 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/spawn-zone. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.521.4 File**

Defined in file src/infinity/legacy-commands.lisp.



## 8.522 Tootsville::Endpoint-Post-/ World/ Infinity/ Speak $\mapsto$ Json

### 8.522.1 Function

Endpoint-Post-/ World/ Infinity/ Speak $\mapsto$ Json names a function, with lambda list NIL:

The user speaks SPEECH at volume VOL in public.

### 8.522.2 Infinity Mode command

See Section 8.736 [TOOTSVILLE INFINITY-SPEAK], page 1016,

### 8.522.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/speak. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.522.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.523 Tootsville::Endpoint-Post-/ World/ Infinity/ Stamp-Passport $\mapsto$ Json

### 8.523.1 Function

Endpoint-Post-/ World/ Infinity/ Stamp-Passport $\mapsto$ Json names a function, with lambda list NIL:

Stamp the Toot's passport

### 8.523.2 Infinity Mode command

See Section 8.737 [TOOTSVILLE INFINITY-STAMP-PASSPORT], page 1019,

### 8.523.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/stamp-passport. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.523.4 File

Defined in file src/infinity/tootsville-commands.lisp.

## 8.524 Tootsville::Endpoint-Post-/ World/ Infinity/ Start-Event $\mapsto$ Json

### 8.524.1 Function

Endpoint-Post-/ World/ Infinity/ Start-Event $\mapsto$ Json names a function, with lambda list NIL:

Attempt to begin a Quaestor Event. Might return an error.

### 8.524.2 Infinity Mode command

See Section 8.738 [TOOTSVILLE INFINITY-START-EVENT], page 1020,

### 8.524.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/start-event. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.524.4 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.525 Tootsville::Endpoint-Post-/ World/ Infinity/ Toot-List $\mapsto$ Json**

### **8.525.1 Function**

Endpoint-Post-/ World/ Infinity/ Toot-List $\mapsto$ Json names a function, with lambda list NIL:  
Enumerates all Toots owned by the user.

### **8.525.2 Infinity Mode command**

See Section 8.740 [TOOTSVILLE INFINITY-TOOT-LIST], page 1024,

### **8.525.3 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/toot-list. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.525.4 File**

Defined in file src/infinity/new-commands-20.lisp.

## 8.526 Tootsville::Endpoint-Post-/ World/ Infinity/ Use-Equipment $\mapsto$ Json

### 8.526.1 Function

Endpoint-Post-/ World/ Infinity/ Use-Equipment $\mapsto$ Json names a function, with lambda list NIL:

The player wishes to use a piece of equipment on a particular item or place.

### 8.526.2 Infinity Mode command

See Section 8.741 [TOOTSVILLE INFINITY-USE-EQUIPMENT], page 1025,

### 8.526.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/use-equipment. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.526.4 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.527 Tootsville::Endpoint-Post-/ World/ Infinity/ Wardrobe $\mapsto$ Json**

### **8.527.1 Function**

Endpoint-Post-/ World/ Infinity/ Wardrobe $\mapsto$ Json names a function, with lambda list NIL:

Describe what your Toot is wearing.

### **8.527.2 Infinity Mode command**

See Section 8.742 [TOOTSVILLE INFINITY-WARDROBE], page 1026,

### **8.527.3 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/wardrobe. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.527.4 File**

Defined in file src/infinity/new-commands-20.lisp.

## 8.528 Tootsville::Endpoint-Post-/ World/ Infinity/ Wtl-4→Json

### 8.528.1 Function

Endpoint-Post-/ World/ Infinity/ Wtl-4→Json names a function, with lambda list NIL:

Walk the Line indirect refresher from observer

### 8.528.2 Infinity Mode command

See Section 8.745 [TOOTSVILLE INFINITY-WTL-4], page 1030,

### 8.528.3 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/wtl-4. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.528.4 File

Defined in file src/infinity/new-commands-20.lisp.

## **8.529 Tootsville::Endpoint-Post-/ World/ Infinity/ Wtl→Json**

### **8.529.1 Function**

Endpoint-Post-/ World/ Infinity/ Wtl→Json names a function, with lambda list NIL:  
Walk the Line

### **8.529.2 Infinity Mode command**

See Section 8.744 [TOOTSVILLE INFINITY-WTL], page 1028,

### **8.529.3 Web Service Endpoint**

This is a web service endpoint accessed by the HTTP method POST at the URI template /world/infinity/wtl. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### **8.529.4 File**

Defined in file src/infinity/new-commands-20.lisp.



## 8.530 Tootsville::Endpoint-Post-/ World/ Infinity $\mapsto$ Json

### 8.530.1 Function

Endpoint-Post-/ World/ Infinity $\mapsto$ Json names a function, with lambda list NIL:

Dispatch an Infinity-mode JSON packet to its handler based on the `c` parameter.

See Section 8.338 [TOOTSVILLE DEFINFINITY], page 592, for a detailed discussion of this mode of operation.

### 8.530.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method POST at the URI template `/world/infinity`. It returns a content-type of `application/json`.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.530.3 File

Defined in file `src/infinity/infinity.lisp`.

## 8.531 Tootsville::Endpoint-Put-/ Toots/ Toot-Name→Json

### 8.531.1 Function

Endpoint-Put-/ Toots/ Toot-Name→Json names a function, with lambda list (TOOT-NAME):

Set properties of a Toot. Currently only child-code.

### 8.531.2 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method PUT at the URI template /toots/:toot-name. It returns a content-type of application/json.

TOOT-NAME is a parameter from the URI.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.531.3 File

Defined in file src/endpoints/slash-toots.lisp.

## 8.532 Tootsville::Endpoint-Put-/ Users/ Me→Json

### 8.532.1 Function

Endpoint-Put-/ Users/ Me→Json names a function, with lambda list NIL:

Makes changes to an user account.

Input JSO:

```
{ key: "field", newValue: "x" }
```

Fields and value formats:

`displayName` (`fullName`)

`givenName`

`surname` (`familyName`)

`sensitive` (`sensitiveP`)

Must be "true" or "false" (as a string)

`lang` (`language`)

Must be a supported ISO language string; e.g. "en-US"

`gender` Must be one of "" "" "", where "" is gender-neutral. Selects pronouns; respectively, "they," "she," or "he."

`dob` (`dateOfBirth`)

Format in RFC-3339 timestamp format; eg, "1990-05-21T00:00:00-0400" or just "1990-05-21"

### 8.532.2 Status: 201 Created

XXX is there a better status for updates?

### 8.532.3 Status: 401 Authorization Required

### 8.532.4 Status: 403 Authorization Failed

### 8.532.5 Status: 405 Not Allowed

### 8.532.6 Status: 422

### 8.532.7 Web Service Endpoint

This is a web service endpoint accessed by the HTTP method PUT at the URI template /users/me. It returns a content-type of application/json.

There are no URI parameters.

It will report a slow response if it takes longer than 0.03 seconds (30 milliseconds) to complete.

### 8.532.8 File

Defined in file src/endpoints/slash-users.lisp.

## **8.533 Tootsville::Endpoint-Template**

### **8.533.1 Function**

Endpoint-Template names an undocumented function, with lambda list (OBJECT).

## **8.534 Tootsville::Endpoint-Template-Arity**

### **8.534.1 Function**

Endpoint-Template-Arity names an undocumented function, with lambda list (OBJECT).

## **8.535 Tootsville::Endpoint-Template-Match**

### **8.535.1 Function**

Endpoint-Template-Match names an undocumented function, with lambda list (ENDPOINT URI-PARTS).

### **8.535.2 File**

Defined in file src/endpoint.lisp.

## 8.536 Tootsville::Endpoint-Template-String

### 8.536.1 Function

Endpoint-Template-String names an undocumented function, with lambda list (ENDPOINT).

### 8.536.2 File

Defined in file src/endpoint.lisp.

## **8.537 Tootsville::Endpoint-Vars->Openapi**

### **8.537.1 Function**

Endpoint-Vars->Openapi names an undocumented function, with lambda list (ENDPOINT).

### **8.537.2 File**

Defined in file src/endpoints/slash-meta-game.lisp.



## 8.538 Tootsville::Endpoints-Equal

### 8.538.1 Function

Endpoints-Equal names a function, with lambda list (A B):

Are A and B references to the identical endpoint URI pattern?

Note that URIs that are not ENDPOINTS-EQUAL to one another can still conflict with one another in URI space. A template could have a variable term which differs from the matching term (URI path element) in the other template, but creates an ambiguity between them (both could plausibly accept some subset of matching URIs). The simplest form is something like: `‘/a/:x’` cv. `‘/a/b’` — it is perfectly possible that `‘:x’` could be `‘b’`, making `‘/a/b’` ambiguous between the two URIs.

There are two possible cures for this bug; let’s say, “avoidance” and “CLOS.” With the CLOS solution, the more specific (less variables) method would override, just as a more specific method overrides a less specific method in the default method combination method in CLOS. The alternative is to not permit such URI pairs to exist at all.

Neither solution has yet been implemented.

### 8.538.2 File

Defined in file `src/endpoint.lisp`.

## **8.539 Tootsville::Endpoints-Page-Footer**

### **8.539.1 Function**

Endpoints-Page-Footer names an undocumented function, with lambda list NIL.

### **8.539.2 File**

Defined in file `src/endpoints/slash-meta-game.lisp`.

## **8.540 Tootsville::Endpoints-Page-Header**

### **8.540.1 Function**

Endpoints-Page-Header names an undocumented function, with lambda list NIL.

### **8.540.2 File**

Defined in file `src/endpoints/slash-meta-game.lisp`.

## **8.541 Tootsville::Endpoints-Prefixed**

### **8.541.1 Function**

Endpoints-Prefixed names an undocumented function, with lambda list (ENDPOINTS).

### **8.541.2 File**

Defined in file `src/endpoints/slash-meta-game.lisp`.

## 8.542 Tootsville::Ensure-Integer

### 8.542.1 Function

Ensure-Integer names a function, with lambda list (VALUE):

Ensure that VALUE is an integer.

Parse strings using PARSE-INTEGGER (see the Common Lisp HyperSpec).

Round real numbers.

### 8.542.2 File

Defined in file src/types/binary.lisp.

## **8.543 Tootsville::Ensure-Inventory-Item**

### **8.543.1 Function**

Ensure-Inventory-Item names an undocumented function, with lambda list (ITEM).

## 8.544 Tootsville::Ensure-List-Of-People

### 8.544.1 Function

Ensure-List-Of-People names a function, with lambda list (IDENTIFIER):

Map IDENTIFIER to a list of humans.

IDENTIFIER may be:

- A person
- A Toot (whose owner is returned)
- A person's eMail address
- A Toot name
- A person or Toot's UUID, in UUID or string-UUID form
- A list of any of the above
- A string list of the above, joined by #, or #;
- NIL

### 8.544.2 File

Defined in file src/staff-journal.lisp.

## **8.545 Tootsville::Ensure-Message-Is-Characters**

### **8.545.1 Function**

Ensure-Message-Is-Characters names a function, with lambda list (MESSAGE):

Convert MESSAGE into a string of characters, probably as JSON.

### **8.545.2 File**

Defined in file src/websockets.lisp.



## 8.546 Tootsville::Ensure-Number

### 8.546.1 Function

Ensure-Number names a function, with lambda list (VALUE):

Ensure that VALUE is a number.

Parse strings using `ORG.MAPCAR.PARSE-NUMBER::PARSE-NUMBER` (not in this manual).

### 8.546.2 File

Defined in file `src/types/binary.lisp`.

## 8.547 Tootsville::Ensure-Record

### 8.547.1 Function

Ensure-Record names an undocumented function, with lambda list (TYPE &REST COLUMNS+VALUES).

### 8.547.2 File

Defined in file src/db/db-central.lisp.

## **8.548 Tootsville::Ensure-Site-Name**

### **8.548.1 Function**

Ensure-Site-Name names an undocumented function, with lambda list NIL.

### **8.548.2 File**

Defined in file src/version.lisp.

## **8.549 Tootsville::Ensure-Toot**

### **8.549.1 Function**

Ensure-Toot names an undocumented function, with lambda list (TOOT).

### **8.549.2 File**

Defined in file src/toots.lisp.

## 8.550 Tootsville::Ensure-User-For-Plist

### 8.550.1 Function

Ensure-User-For-Plist names a function, with lambda list (PLIST):

Find or create the user described by PLIST and return them.

PLIST can have keys that align to a DB.PERSON or their contact infos (eg, email) and is expected to have been validated already (eg, come from a trusted authentication provider like Google Firebase).

### 8.550.2 File

Defined in file src/users.lisp.

## **8.551 Tootsville::Ensure-Wear-Slot**

### **8.551.1 Function**

Ensure-Wear-Slot names an undocumented function, with lambda list (SLOT).

## **8.552 Tootsville::Ensure-Weather-Kernel**

### **8.552.1 Function**

Ensure-Weather-Kernel names an undocumented function, with lambda list NIL.

### **8.552.2 File**

Defined in file `src/weather/weather.lisp`.

## 8.553 Tootsville::Entry

### 8.553.1 Function

Entry names a function, with lambda list (&OPTIONAL ARGV):

Top-level entry-point for the compiled executable binary form.

Dispatches based upon the single argument, expected to be a verb (case-insensitive) from the hard-coded table in this function.

### 8.553.2 File

Defined in file `src/command-line.lisp`.



## **8.554 Tootsville::Enumerate-Endpoints**

### **8.554.1 Function**

Enumerate-Endpoints names an undocumented function, with lambda list NIL.

### **8.554.2 File**

Defined in file `src/endpoints/slash-meta-game.lisp`.

## **8.555 Tootsville::Erase-All-Memcached-For**

### **8.555.1 Function**

Erase-All-Memcached-For names an undocumented function, with lambda list (NAME &REST COLUMNS+VALUES).

### **8.555.2 File**

Defined in file src/db/memcached.lisp.

## **8.556 Tootsville::Error-Log-File**

### **8.556.1 Function**

Error-Log-File names a function, with lambda list (LOG-DIR):

Get the pathname of the error log file.

### **8.556.2 File**

Defined in file src/logging.lisp.

## **8.557 Tootsville::Every-Toot-Name**

### **8.557.1 Function**

Every-Toot-Name names a function, with lambda list NIL:

Enumerates the names of every Toot known to the system.

### **8.557.2 File**

Defined in file src/toots.lisp.

## **8.558 Tootsville::Extension-For-Content-Type**

### **8.558.1 Function**

Extension-For-Content-Type names a function, with lambda list (CONTENT-TYPE):

Get the canonically-preferred filename extension for CONTENT-TYPE.

### **8.558.2 File**

Defined in file src/web.lisp.

## **8.559 Tootsville::Extract-Certificate-Base64**

### **8.559.1 Function**

Extract-Certificate-Base64 names a function, with lambda list (STRING):

Base64-decode the certificate in *STRING* between *BEGIN CERTIFICATE* header and *END CERTIFICATE* footer lines.

### **8.559.2 File**

Defined in file `src/auth/auth-firebase.lisp`.

## 8.560 Tootsville::Extract-Plist-Path

### 8.560.1 Function

Extract-Plist-Path names an undocumented function, with lambda list (PATH PLIST &OPTIONAL PREFIX).

### 8.560.2 File

Defined in file src/version.lisp.

## **8.561 Tootsville::Extract-Public-Key-From-Cert**

### **8.561.1 Function**

Extract-Public-Key-From-Cert names a function, with lambda list (CERT):

Extract the public key from an X.509 certificate

### **8.561.2 File**

Defined in file `src/endpoints/gossip/alex/alex.lisp`.



## **8.562 Tootsville::Fetch-Ice-Credentials/ Xirsys**

### **8.562.1 Function**

Fetch-Ice-Credentials/Xirsys names an undocumented function, with lambda list NIL.

### **8.562.2 File**

Defined in file src/gossip.lisp.

## **8.563 Tootsville::Fetch-Json**

### **8.563.1 Function**

Fetch-Json names a function, with lambda list (URI):

Fetch URI as an application/json file and parse it with Yason into a property list tree.

### **8.563.2 File**

Defined in file src/browser.lisp.

## **8.564 Tootsville::Fill-Blank-Contour**

### **8.564.1 Function**

Fill-Blank-Contour names an undocumented function, with lambda list (LATITUDE LONGITUDE BASE-ELEVATION).

### **8.564.2 File**

Defined in file src/terrain.lisp.

## **8.565 Tootsville::Find-Acceptor**

### **8.565.1 Function**

Find-Acceptor names a function, with lambda list (HOST PORT):

Find an active Acceptor running on the given HOST address and PORT

### **8.565.2 File**

Defined in file src/main.lisp.

## **8.566 Tootsville::Find-Best-Endpoint**

### **8.566.1 Function**

Find-Best-Endpoint names an undocumented function, with lambda list NIL.

### **8.566.2 File**

Defined in file src/endpoint.lisp.

## **8.567 Tootsville::Find-Client-For-Socket**

### **8.567.1 Function**

Find-Client-For-Socket names an undocumented function, with lambda list (SOCKET).

### **8.567.2 File**

Defined in file src/tcp-stream.lisp.

## **8.568 Tootsville::Find-Exact-Endpoint**

### **8.568.1 Function**

Find-Exact-Endpoint names an undocumented function, with lambda list NIL.

### **8.568.2 File**

Defined in file src/endpoint.lisp.

## **8.569 Tootsville::Find-Infinity-Websocket-Resource**

### **8.569.1 Function**

Find-Infinity-Websocket-Resource names an undocumented function, with lambda list (REQUEST).

### **8.569.2 File**

Defined in file src/websockets.lisp.



## **8.570 Tootsville::Find-Kinda-Endpoint**

### **8.570.1 Function**

Find-Kinda-Endpoint names an undocumented function, with lambda list NIL.

### **8.570.2 File**

Defined in file src/endpoint.lisp.

## **8.571 Tootsville::Find-Log-Dir**

### **8.571.1 Function**

Find-Log-Dir names a function, with lambda list NIL:

Find the logging directory under USER-HOMEDIR-PATHNAME (see the Common Lisp HyperSpec)

### **8.571.2 File**

Defined in file src/logging.lisp.

## 8.572 Tootsville::Find-Person-By-Url

### 8.572.1 Function

Find-Person-By-Url names an undocumented function, with lambda list (URL &OPTIONAL MORE).

### 8.572.2 File

Defined in file src/users.lisp.

## **8.573 Tootsville::Find-Player-Or-Die**

### **8.573.1 Function**

Find-Player-Or-Die names a function, with lambda list NIL:

Ensure that a recognized player is connected.

### **8.573.2 File**

Defined in file src/users.lisp.

## 8.574 Tootsville::Find-Random-Point-If

### 8.574.1 Function

Find-Random-Point-If names a function, with lambda list (FUNCTION):

Find a random point within the space for which FUNCTION is true.

Returns (LIST LATITUDE LONGITUDE)

### 8.574.2 File

Defined in file src/terrain.lisp.

## 8.575 Tootsville::Find-Record

### 8.575.1 Function

Find-Record names a function, with lambda list (TYPE &REST COLUMNS+VALUES):

Find a record of TYPE where each of COLUMNS+VALUES are exact matches.

Expects to find 0 or 1 result. If more results are found, signals an error.

See Section 8.576 [TOOTSVILLE FIND-RECORDS], page 833, for more details.

### 8.575.2 File

Defined in file src/db/generic-db.lisp.

## 8.576 Tootsville::Find-Records

### 8.576.1 Function

Find-Records names a function, with lambda list (TYPE &REST COLUMNS+VALUES):

Find all records of TYPE where each of COLUMNS+VALUES are exact matches.

For each of the columns named, the value given must be an exact match. In the case of SQL, this translates neatly into a construction such as “WHERE column = value, AND column = value, . . . AND column = value.” With other kinds of database (e.g. LDAP, Couch, &c) the equivalent constructions will be used.

This method is not suitable for inequalities, set comparisons, or the like — in fact, only value-like equality is supported.

The function returns NIL if no records are found.

### 8.576.2 File

Defined in file src/db/generic-db.lisp.

## **8.577 Tootsville::Find-Records-By-Sql**

### **8.577.1 Function**

Find-Records-By-Sql names a function, with lambda list (TYPE SQL):

Find records of type TYPE by using the query SQL.

SQL must be a query of the form “SELECT \* FROM table WHERE. . .”

### **8.577.2 File**

Defined in file src/db/generic-db.lisp.



## 8.578 Tootsville::Find-Reference

### 8.578.1 Function

Find-Reference names a function, with lambda list (OBJECT FIELD):

Following the FIELD on OBJECT, return the referenced object.

Note that this returns an object of the appropriate type, not its ID code. The regular column reference function (CLASS)-(FIELD) will return the ID value, which may be of any type (eg, UUID, STRING, NUMBER, &c)

### 8.578.2 File

Defined in file src/db/generic-db.lisp.

## **8.579 Tootsville::Find-Results-In-Docstring**

### **8.579.1 Function**

Find-Results-In-Docstring names an undocumented function, with lambda list (DOCSTRING).

### **8.579.2 File**

Defined in file src/endpoints/slash-meta-game.lisp.

## 8.580 Tootsville::Find-Robot

### 8.580.1 Function

Find-Robot names a function, with lambda list (IDENTIFIER):

Find a robot based on IDENTIFIER.

IDENTIFIER may be a name string or Toot object.

### 8.580.2 File

Defined in file src/characters/robots.lisp.

## **8.581 Tootsville::Find-Thread**

### **8.581.1 Function**

Find-Thread names a function, with lambda list (NAME):

Find any thread whose name includes NAME

### **8.581.2 File**

Defined in file src/messaging.lisp.

## **8.582 Tootsville::Find-Toot-By-Name**

### **8.582.1 Function**

Find-Toot-By-Name names an undocumented function, with lambda list (TOOT-NAME).

### **8.582.2 File**

Defined in file src/toots.lisp.

## **8.583 Tootsville::Find-Toot-Passport**

### **8.583.1 Function**

Find-Toot-Passport names a function, with lambda list (TOOT):

Return the passport for TOOT

### **8.583.2 File**

Defined in file src/passport.lisp.

## 8.584 Tootsville::Find-User-For-Credentials

### 8.584.1 Function

Find-User-For-Credentials names an undocumented function, with lambda list (CREDENTIALS).

### 8.584.2 File

Defined in file src/users.lisp.

## **8.585 Tootsville::Find-User-For-Headers**

### **8.585.1 Function**

Find-User-For-Headers names an undocumented function, with lambda list (STRING).

### **8.585.2 File**

Defined in file src/acceptor.lisp.



## **8.586 Tootsville::Find-User-For-Json**

### **8.586.1 Function**

Find-User-For-Json names a function, with lambda list (JSON):

Find a user based on submitted authentication JSON

### **8.586.2 File**

Defined in file src/websockets.lisp.

## **8.587 Tootsville::Find-Var-In-Docstring**

### **8.587.1 Function**

Find-Var-In-Docstring names an undocumented function, with lambda list (VARIABLE DOCSTRING).

### **8.587.2 File**

Defined in file `src/endpoints/slash-meta-game.lisp`.

## **8.588 Tootsville::First-Line**

### **8.588.1 Function**

First-Line names a function, with lambda list (STRING):

The first line, or, up to 100 characters of STRING.

### **8.588.2 File**

Defined in file src/web.lisp.

## **8.589 Tootsville::First-Paragraph**

### **8.589.1 Function**

First-Paragraph names a function, with lambda list (STRING):

Returns the first paragraph of STRING.

(Up to the first blank line)

### **8.589.2 File**

Defined in file src/types/string-characteristics.lisp.

## 8.590 Tootsville::Flatten-Plist-Tree

### 8.590.1 Function

Flatten-Plist-Tree names an undocumented function, with lambda list (NODE &OPTIONAL (PREFIX )).

### 8.590.2 File

Defined in file src/http-error.lisp.

## **8.591 Tootsville::Flora-Personality**

### **8.591.1 Class**

Flora-Personality names a class, with one superclass: Section 8.1080 [TOOTSVILLE ROBOT-FLORA], page 1365.

This class defines a character named Flora

### **8.591.2 Slots**

Class Flora-Personality has no direct slots defined.

## 8.592 Tootsville::Force-Close-Hunchensocket

### 8.592.1 Function

Force-Close-Hunchensocket names a function, with lambda list (CLIENT):

Attempt to destroy the connection to CLIENT

### 8.592.2 File

Defined in file src/websockets.lisp.

## **8.593 Tootsville::Fountain-Duplicate-P**

### **8.593.1 Function**

Fountain-Duplicate-P names a function, with lambda list (EVENT-SOURCE):

Returns generalized true if EVENT-SOURCE has happened already on the same Tootsville day as today.

### **8.593.2 File**

Defined in file src/quaestor.lisp.



## **8.594 Tootsville::Fountain-Reject-As-Already-Done**

### **8.594.1 Function**

Fountain-Reject-As-Already-Done names a function, with lambda list (MONIKER):

Send a rejection to an attempt to end a fountain event identified by MONIKER.

Tells the player to make a wish again tomorrow.

### **8.594.2 File**

Defined in file src/quaestor.lisp.

## 8.595 Tootsville::From-Avatars

### 8.595.1 Function

From-Avatars names a function, with lambda list (TOOTS-WITH-KEYS):

Returns a from: "avatars" packet which is the result of a number of commands.

The packet format is as follows:

```
{ from: "avatars",  
  avatars: { KEY: TOOT-INFO, [ ... ] },  
  inRoom: "@Tootsville",  
  status: true }
```

The avatar information is in the form given by Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534, q.v.

The parameter TOOTS-WITH-KEYS is a property list whose keys are arbitrary strings (or symbols, whose names will be taken) and whose values are Toot designators suitable to be passed to Section 8.549 [TOOTSVILLE ENSURE-TOOT], page 806, eg. Toot names or Toot objects.

### 8.595.2 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.596 Tootsville::Game-Action-Bowling-Reset-Pins

### 8.596.1 Function

Game-Action-Bowling-Reset-Pins names a function, with lambda list (ACTION):

Reset the pins and move to the next player or frame of bowling.

### 8.596.2 Usage

WRITEME

### 8.596.3 Effects

WRITEME WRITEME

See Section 8.612 [TOOTSVILLE GAME-ACTION-START-BOWLING], page 870, for an overview of bowling.

### 8.596.4 File

Defined in file `src/infinity/game-actions.lisp`.

## **8.597 Tootsville::Game-Action-Bowling-Strike-Pins**

### **8.597.1 Function**

Game-Action-Bowling-Strike-Pins names a function, with lambda list (ACTION):

Record the bowling ball striking the pins

### **8.597.2 Usage**

WRITEME

### **8.597.3 Effects**

WRITEME WRITEME

See Section 8.612 [TOOTSVILLE GAME-ACTION-START-BOWLING], page 870, for an overview of bowling.

### **8.597.4 File**

Defined in file `src/infinity/game-actions.lisp`.

## 8.598 Tootsville::Game-Action-Card-Game-Arrange

### 8.598.1 Function

Game-Action-Card-Game-Arrange names a function, with lambda list (ACTION):

Re-order the cards in your hand.

### 8.598.2 Usage

WRITEME

### 8.598.3 Effects

WRITEME WRITEME

See Section 8.607 [TOOTSVILLE GAME-ACTION-JOIN-CARD-GAME], page 864, for an overview of card table games.

### 8.598.4 File

Defined in file src/infinity/game-actions.lisp.

## 8.599 Tootsville::Game-Action-Card-Game-Deal

### 8.599.1 Function

Game-Action-Card-Game-Deal names a function, with lambda list (ACTION):

Deal a card from the shuffled deck to another player without looking.

### 8.599.2 Usage

WRITEME

### 8.599.3 Effects

WRITEME WRITEME

See Section 8.607 [TOOTSVILLE GAME-ACTION-JOIN-CARD-GAME], page 864, for an overview of card table games.

### 8.599.4 File

Defined in file `src/infinity/game-actions.lisp`.

## 8.600 Tootsville::Game-Action-Card-Game-Draw

### 8.600.1 Function

Game-Action-Card-Game-Draw names a function, with lambda list (ACTION):

Draw a card from the shuffled deck into your hand.

### 8.600.2 Usage

WRITEME

### 8.600.3 Effects

WRITEME WRITEME

See Section 8.607 [TOOTSVILLE GAME-ACTION-JOIN-CARD-GAME], page 864, for an overview of card table games.

### 8.600.4 File

Defined in file `src/infinity/game-actions.lisp`.

## **8.601 Tootsville::Game-Action-Card-Game-Move**

### **8.601.1 Function**

Game-Action-Card-Game-Move names a function, with lambda list (ACTION):

Move a card around on the table.

### **8.601.2 Usage**

WRITEME

### **8.601.3 Effects**

WRITEME WRITEME

See Section 8.607 [TOOTSVILLE GAME-ACTION-JOIN-CARD-GAME], page 864, for an overview of card table games.

### **8.601.4 File**

Defined in file `src/infinity/game-actions.lisp`.



## 8.602 Tootsville::Game-Action-Card-Game-Play

### 8.602.1 Function

Game-Action-Card-Game-Play names a function, with lambda list (ACTION):

Play a card from your hand, placing it on the table.

The card may be played face-down or face-up, and at any place on the table.

### 8.602.2 Usage

WRITEME

### 8.602.3 Effects

WRITEME

WRITEME

See Section 8.607 [TOOTSVILLE GAME-ACTION-JOIN-CARD-GAME], page 864, for an overview of card table games.

### 8.602.4 File

Defined in file src/infinity/game-actions.lisp.

## 8.603 Tootsville::Game-Action-Card-Game-Shuffle

### 8.603.1 Function

Game-Action-Card-Game-Shuffle names a function, with lambda list (ACTION):

Shuffle all cards into the deck.

### 8.603.2 Usage

WRITEME

### 8.603.3 Effects

WRITEME WRITEME

See Section 8.607 [TOOTSVILLE GAME-ACTION-JOIN-CARD-GAME], page 864, for an overview of card table games.

### 8.603.4 File

Defined in file src/infinity/game-actions.lisp.

## 8.604 Tootsville::Game-Action-Card-Game-Take

### 8.604.1 Function

Game-Action-Card-Game-Take names a function, with lambda list (ACTION):

Pick up a card from the table, placing it into your hand.

### 8.604.2 Usage

WRITEME

### 8.604.3 Effects

WRITEME WRITEME

See Section 8.607 [TOOTSVILLE GAME-ACTION-JOIN-CARD-GAME], page 864, for an overview of card table games.

### 8.604.4 File

Defined in file `src/infinity/game-actions.lisp`.

## 8.605 Tootsville::Game-Action-Get-Bowling-Scorecard

### 8.605.1 Function

Game-Action-Get-Bowling-Scorecard names a function, with lambda list (ACTION):

Get the scorecard for a bowling game in progress.

### 8.605.2 Usage

WRITEME

### 8.605.3 Effects

WRITEME WRITEME

See Section 8.612 [TOOTSVILLE GAME-ACTION-START-BOWLING], page 870, for an overview of bowling.

### 8.605.4 File

Defined in file `src/infinity/game-actions.lisp`.

## 8.606 Tootsville::Game-Action-Join-Bowling-Game

### 8.606.1 Function

Game-Action-Join-Bowling-Game names a function, with lambda list (ACTION):

Join a bowling game that's about to start

### 8.606.2 Usage

WRITEME

### 8.606.3 Effects

WRITEME WRITEME

See Section 8.612 [TOOTSVILLE GAME-ACTION-START-BOWLING], page 870, for an overview of bowling.

### 8.606.4 File

Defined in file `src/infinity/game-actions.lisp`.

## 8.607 Tootsville::Game-Action-Join-Card-Game

### 8.607.1 Function

Game-Action-Join-Card-Game names a function, with lambda list (ACTION):

Start playing a card game.

### 8.607.2 Usage

```
{ c: "gameAction",
  d: { action: "joinCardGame",
      playerP: [ true | false ],
      cardTable: UUID } }
```

### 8.607.3 Overview of Card Games

Playing cards on a card table uses a special camera view to show the table top, and a pop-over layer to show the cards in the local player's hand.

Up to 4 players can join the game. Each player gets a side of the card table. See also Section 8.609 [TOOTSVILLE GAME-ACTION-PART-CARD-GAME], page 867.

The deck of cards can be 52 cards or have the 2 jokers for 54 in total. See Section 8.603 [TOOTSVILLE GAME-ACTION-CARD-GAME-SHUFFLE], page 860.

Players can draw from the deck (Section 8.600 [TOOTSVILLE GAME-ACTION-CARD-GAME-DRAW], page 857), deal cards to other players (Section 8.599 [TOOTSVILLE GAME-ACTION-CARD-GAME-DEAL], page 856), and place cards on the table face-up or face-down (Section 8.602 [TOOTSVILLE GAME-ACTION-CARD-GAME-PLAY], page 859), pick up cards from the table (Section 8.604 [TOOTSVILLE GAME-ACTION-CARD-GAME-TAKE], page 861), or move cards around on the table (Section 8.601 [TOOTSVILLE GAME-ACTION-CARD-GAME-MOVE], page 858) or in your hand (Section 8.598 [TOOTSVILLE GAME-ACTION-CARD-GAME-ARRANGE], page 855).

There are no particular rules of any card game enforced. Players are free to do whatever they like with the cards.

### 8.607.4 Joining a Card Game

The ACTION passed references a card table. If that card table already has 4 players, this player can only become an observer. Otherwise, the player can choose to play or to observe.

The structure of ACTION includes these keys:

**cardTable**

The UUID of a card table.

**playerP** If true, this Toot wants to be a player. If false, this Toot wants to be an observer.

Attempting to join a card game as a fifth player will result in an error.

```
{ from: "gameAction", action: "joinCardGame", status: false,
  error: USER-ERROR-MESSAGE }
```

The user error message will be something suitable for display to the user to explain why they were refused joining the game.

On success, the player receives a datagram such as

```
{ from: "gameAction", action: "joinCardGame", status: true }
```

The player is then able to issue other gameAction packets as described in the preceding overview.

### **8.607.5 Usage**

WRITEME

### **8.607.6 Effects**

WRITEME

### **8.607.7 File**

Defined in file `src/infinity/game-actions.lisp`.

## 8.608 Tootsville::Game-Action-Part-Bowling-Game

### 8.608.1 Function

Game-Action-Part-Bowling-Game names a function, with lambda list (ACTION):

Quit a bowling game that's about to start or already started.

### 8.608.2 Usage

WRITEME

### 8.608.3 Effects

WRITEME WRITEME

See Section 8.612 [TOOTSVILLE GAME-ACTION-START-BOWLING], page 870, for an overview of bowling.

### 8.608.4 File

Defined in file `src/infinity/game-actions.lisp`.



## 8.609 Tootsville::Game-Action-Part-Card-Game

### 8.609.1 Function

Game-Action-Part-Card-Game names a function, with lambda list (ACTION):

Quit a card game at a card table.

### 8.609.2 Usage

WRITEME

### 8.609.3 Effects

WRITEME WRITEME

See Section 8.607 [TOOTSVILLE GAME-ACTION-JOIN-CARD-GAME], page 864, for an overview of card table games.

### 8.609.4 File

Defined in file src/infinity/game-actions.lisp.

## 8.610 Tootsville::Game-Action-Pause-Sports-Ball-Timer

### 8.610.1 Function

Game-Action-Pause-Sports-Ball-Timer names a function, with lambda list (ACTION):

Pause the timer for a SportsBall game.

### 8.610.2 Usage

WRITEME

### 8.610.3 Effects

WRITEME

See: Section 8.613 [TOOTSVILLE GAME-ACTION-START-SPORTS-BALL-GAME], page 871, Section 8.614 [TOOTSVILLE GAME-ACTION-START-SPORTS-BALL-TIMER], page 872,

### 8.610.4 File

Defined in file src/infinity/game-actions.lisp.

## 8.611 Tootsville::Game-Action-Sports-Ball-Goal

### 8.611.1 Function

Game-Action-Sports-Ball-Goal names a function, with lambda list (ACTION):

Score a goal in a SportsBall game.

### 8.611.2 Usage

WRITEME

### 8.611.3 Example

WRITEME

### 8.611.4 Effects

WRITEME

See: Section 8.613 [TOOTSVILLE GAME-ACTION-START-SPORTS-BALL-GAME], page 871,

### 8.611.5 File

Defined in file src/infinity/game-actions.lisp.

## 8.612 Tootsville::Game-Action-Start-Bowling

### 8.612.1 Function

Game-Action-Start-Bowling names a function, with lambda list (ACTION):

Start a bowling game.

This action takes a bowling lane as an argument. The lane is reset and the scoreboard is wiped clear.

### 8.612.2 Usage

WRITEME

### 8.612.3 Effects

WRITEME

### 8.612.4 Overview of Bowling

WRITEME

### 8.612.5 Bowling gameAction actions

- Section 8.596 [TOOTSVILLE GAME-ACTION-BOWLING-RESET-PINS], page 853,
- Section 8.597 [TOOTSVILLE GAME-ACTION-BOWLING-STRIKE-PINS], page 854,
- Section 8.606 [TOOTSVILLE GAME-ACTION-JOIN-BOWLING-GAME], page 863,
- Section 8.608 [TOOTSVILLE GAME-ACTION-PART-BOWLING-GAME], page 866,
- Section 8.605 [TOOTSVILLE GAME-ACTION-GET-BOWLING-SCORECARD], page 862,

### 8.612.6 Starting a Bowling Game

WRITEME

### 8.612.7 File

Defined in file src/infinity/game-actions.lisp.

## 8.613 Tootsville::Game-Action-Start-Sports-Ball-Game

### 8.613.1 Function

Game-Action-Start-Sports-Ball-Game names a function, with lambda list (ACTION):

Start a SportsBall game.

### 8.613.2 Usage

A gameAction packet of the form:

```
{ c: "gameAction",
  d: { action: "startSportsBallGame",
       game: UUID } }
```

### 8.613.3 Example

```
{ c: "gameAction",
  d: { action: "startSportsBallGame",
       game: "AEB967CB-5598-40D5-9B4A-894C9BC38501" } }
```

### 8.613.4 Effects

Sending startSportsBallGame initiates a soccer or other game based on the same basic premise. The score board is initialized to 0 vs. 0 points, and teams spot one another's goals with Section 8.611 [TOOTSVILLE GAME-ACTION-SPORTS-BALL-GOAL], page 869, to increment the score. Optionally, a timer can be started with startSportsBallTimer, see Section 8.614 [TOOTSVILLE GAME-ACTION-START-SPORTS-BALL-TIMER], page 872, Section 8.610 [TOOTSVILLE GAME-ACTION-PAUSE-SPORTS-BALL-TIMER], page 868, 'GAME-ACTION-RESUME-SPORTS-BALL-TIMER'.

### 8.613.5 About SportsBall

SportsBall is a game system that is able to roughly support a football (soccer) match, but does not actively enforce many rules.

In other words, players could choose to play football (soccer) in the game, but they could also "cheat" or ignore rules such as "off sides" that are inconvenient to implement.

The basic principle is that the ball is moving across the field, and should be returned to the sidelines when it goes out of bounds. The players attempt to move the ball into one of two goals. When the game field is dedicated to the game, these goal objects must be designated.

Players can click on the ball to run towards it and kick it. Collisions between the ball and players will bounce the ball through the field.

### 8.613.6 File

Defined in file src/infinity/game-actions.lisp.

## **8.614 Tootsville::Game-Action-Start-Sports-Ball-Timer**

### **8.614.1 Function**

Game-Action-Start-Sports-Ball-Timer names a function, with lambda list (ACTION):

Resume the timer for a SportsBall game after it had been paused.

### **8.614.2 Usage**

WRITEME

### **8.614.3 Effects**

WRITEME WRITEME

See: Section 8.613 [TOOTSVILLE GAME-ACTION-START-SPORTS-BALL-GAME], page 871, Section 8.614 [TOOTSVILLE GAME-ACTION-START-SPORTS-BALL-TIMER], page 872,

### **8.614.4 File**

Defined in file src/infinity/game-actions.lisp.

## **8.615 Tootsville::Game-Action-Tag-You-Re-It**

### **8.615.1 Function**

Game-Action-Tag-You-Re-It names a function, with lambda list (ACTION):

### **8.615.2 Usage**

WRITEME

### **8.615.3 Effects**

WRITEME WRITEME

### **8.615.4 File**

Defined in file src/infinity/game-actions.lisp.

## 8.616 Tootsville::Game-Point

### 8.616.1 Class

Game-Point names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.616.2 Slots

Class Game-Point has 7 direct slot definitions:

Latitude

Longitude

Altitude

World

X

Y

Z



## **8.617 Tootsville::Game-Point-Altitude**

### **8.617.1 Function**

Game-Point-Altitude names an undocumented function, with lambda list (INSTANCE).

### **8.617.2 File**

Defined in file src/characters/robots.lisp.

### **8.617.3 SetF Function**

(SETF Game-Point-Altitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.617.4 File**

Defined in file src/characters/robots.lisp.

## **8.618 Tootsville::Game-Point-Latitude**

### **8.618.1 Function**

Game-Point-Latitude names an undocumented function, with lambda list (INSTANCE).

### **8.618.2 File**

Defined in file src/characters/robots.lisp.

### **8.618.3 SetF Function**

(SETF Game-Point-Latitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.618.4 File**

Defined in file src/characters/robots.lisp.

## 8.619 Tootsville::Game-Point-Longitude

### 8.619.1 Function

Game-Point-Longitude names an undocumented function, with lambda list (INSTANCE).

### 8.619.2 File

Defined in file src/characters/robots.lisp.

### 8.619.3 SetF Function

(SETF Game-Point-Longitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.619.4 File

Defined in file src/characters/robots.lisp.

## 8.620 Tootsville::Game-Point-P

### 8.620.1 Function

Game-Point-P names an undocumented function, with lambda list (OBJECT).

### 8.620.2 File

Defined in file `src/characters/robots.lisp`.

## **8.621 Tootsville::Game-Point-World**

### **8.621.1 Function**

Game-Point-World names an undocumented function, with lambda list (INSTANCE).

### **8.621.2 File**

Defined in file src/characters/robots.lisp.

### **8.621.3 SetF Function**

(SETF Game-Point-World) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.621.4 File**

Defined in file src/characters/robots.lisp.

## **8.622 Tootsville::Game-Point-X**

### **8.622.1 Function**

Game-Point-X names an undocumented function, with lambda list (INSTANCE).

### **8.622.2 File**

Defined in file src/characters/robots.lisp.

### **8.622.3 SetF Function**

(SETF Game-Point-X) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.622.4 File**

Defined in file src/characters/robots.lisp.

## **8.623 Tootsville::Game-Point-Y**

### **8.623.1 Function**

Game-Point-Y names an undocumented function, with lambda list (INSTANCE).

### **8.623.2 File**

Defined in file src/characters/robots.lisp.

### **8.623.3 SetF Function**

(SETF Game-Point-Y) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.623.4 File**

Defined in file src/characters/robots.lisp.

## **8.624 Tootsville::Game-Point-Z**

### **8.624.1 Function**

Game-Point-Z names an undocumented function, with lambda list (INSTANCE).

### **8.624.2 File**

Defined in file src/characters/robots.lisp.

### **8.624.3 SetF Function**

(SETF Game-Point-Z) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.624.4 File**

Defined in file src/characters/robots.lisp.



## 8.625 Tootsville::Gather-All-Symbols

### 8.625.1 Function

Gather-All-Symbols names a function, with lambda list NIL:

Gathers all defined symbols in Section 8.87 [TOOTSVILLE +DOC-PACKAGES+], page 341,

### 8.625.2 File

Defined in file src/write-docs-2.lisp.

## 8.626 Tootsville::Generate-Blank-Contour

### 8.626.1 Function

Generate-Blank-Contour names an undocumented function, with lambda list (9-ELEVATIONS LATITUDE LONGITUDE).

### 8.626.2 File

Defined in file src/terrain.lisp.

## 8.627 Tootsville::Generate-Buddy-List-Signature

### 8.627.1 Function

Generate-Buddy-List-Signature names an undocumented function, with lambda list (REQUESTOR REQUESTEE).

### 8.627.2 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.628 Tootsville::Generate-Skydome-Cloud-Layer**

### **8.628.1 Function**

Generate-Skydome-Cloud-Layer names an undocumented function, with lambda list NIL.

### **8.628.2 File**

Defined in file src/weather/weather.lisp.

## 8.629 Tootsville::Generate-Terrain-Blank-Edge-Horz

### 8.629.1 Function

Generate-Terrain-Blank-Edge-Horz names an undocumented function, with lambda list (START-LATITUDE LONGITUDE END-LATITUDE BASE-ELEVATION).

### 8.629.2 File

Defined in file src/terrain.lisp.

## **8.630 Tootsville::Generate-Terrain-Blank-Edge-Vert**

### **8.630.1 Function**

Generate-Terrain-Blank-Edge-Vert names an undocumented function, with lambda list (LATITUDE START-LONGITUDE END-LONGITUDE BASE-ELEVATION).

### **8.630.2 File**

Defined in file src/terrain.lisp.

## 8.631 Tootsville::Generate-Terrain-Contour

### 8.631.1 Function

Generate-Terrain-Contour names a function, with lambda list (9-ELEVATIONS HABITAT LATITUDE LONGITUDE SCALE):

Generate the contour for a tile area

### 8.631.2 File

Defined in file src/terrain.lisp.

## 8.632 Tootsville::Generate-Terrain-Features

### 8.632.1 Function

Generate-Terrain-Features names a function, with lambda list (CONTOUR HABITAT):

Generate the terrain features based upon the contour map and habitat type. Methods of this function specialize upon the habitat type.

### 8.632.2 File

Defined in file src/terrain.lisp.



## 8.633 Tootsville::Get-9-Terrain-Tiles

### 8.633.1 Function

Get-9-Terrain-Tiles names a function, with lambda list (LATITUDE LONGITUDE):

Returns 9 tiles of terrain centered on LATITUDE LONGITUDE as a 3 by 3 array

### 8.633.2 File

Defined in file src/terrain.lisp.

## **8.634 Tootsville::Get-Google-Account-Keys**

### **8.634.1 Function**

Get-Google-Account-Keys names an undocumented function, with lambda list NIL.

### **8.634.2 File**

Defined in file `src/auth/auth-firebase.lisp`.

## **8.635 Tootsville::Get-Last-Insert-Id**

### **8.635.1 Function**

Get-Last-Insert-Id names an undocumented function, with lambda list NIL.

### **8.635.2 File**

Defined in file src/db/db-central.lisp.

## 8.636 Tootsville::Get-Mariadb-Lock

### 8.636.1 Function

Get-Mariadb-Lock names a function, with lambda list (LOCK-STRING &KEY IF-NOT-LOCKED TIMEOUT):

Obtain database lock LOCK-STRING.

See Section 8.1358 [TOOTSVILLE WITH-CLUSTER-WIDE-LOCK-HELD], page 1645, for a practical use of this.

LOCK-STRING is passed to the MariaDB server and a global lock by that name is obtained via MySQL function GET\_LOCK(STRING), if possible.

If the lock is busy, IF-NOT-LOCKED determines the next action.

**:WAIT**

Wait for up to TIMEOUT seconds for the lock to be freed. If the lock cannot be obtained within TIMEOUT seconds, signal an error of type CLUSTER-WIDE-LOCK-BUSY-ERROR. If TIMEOUT is NIL, wait indefinitely until the lock can be obtained.

**:SKIP**

Skip BODY and return NIL.

**:WARN**

Signal a warning of type CLUSTER-WIDE-LOCK-BUSY-WARNING, then skip BODY and return NIL.

**:ERROR**

Signal an error of type CLUSTER-WIDE-LOCK-BUSY-ERROR.

Returns an opaque identifier that can be passed to ‘YIELD-DB-LOCK’ to release the lock.

LOCK-NAME is case-insensitive.

### 8.636.2 File

Defined in file src/db/maria.lisp.

## 8.637 Tootsville::Get-Rollbar-Person

### 8.637.1 Function

Get-Rollbar-Person names a function, with lambda list (&OPTIONAL (PERSON \*USER\*)):

Return PERSON information for Rollbar error reporting

### 8.637.2 File

Defined in file src/users.lisp.

## 8.638 Tootsville::Get-Unix-Time

### 8.638.1 Function

Get-Unix-Time names a function, with lambda list (&OPTIONAL (UNIVERSAL-TIME (GET-UNIVERSAL-TIME))):

Get the UNIVERSAL-TIME (default to now) in Unix time.

Returns the number of seconds since the Unix epoch, 1970-01-01 at 00:00 Z time.

Note that Java time is Unix time multiplied by 1,000 (ie. time in msec) and is used in some places in the Infinity mode communications.

### 8.638.2 File

Defined in file src/types/date+time.lisp.

## **8.639 Tootsville::Gift-Item**

### **8.639.1 Function**

Gift-Item names a function, with lambda list (ITEM GIVER RECIPIENT):

Transfer the ownership of ITEM from GIVER to RECIPIENT.

### **8.639.2 File**

Defined in file src/items.lisp.

## 8.640 Tootsville::Global-Heightmap-Corner

### 8.640.1 Function

Global-Heightmap-Corner names an undocumented function, with lambda list (LATITUDE LONGITUDE).

### 8.640.2 File

Defined in file src/terrain.lisp.

### 8.640.3 SetF Function

(SETF Global-Heightmap-Corner) names an undocumented function, with lambda list (ELEVATION LATITUDE LONGITUDE).

### 8.640.4 File

Defined in file src/terrain.lisp.



## 8.641 Tootsville::Gone

### 8.641.1 Class

Gone names a class, with one superclass: Section 8.930 [TOOTSVILLE NOT-FOUND], page 1215.

A resource is no longer available.

In particular, this is returned for functions which were discontinued in Romance 2 but existed in earlier versions of the protocol.

### 8.641.2 Slots

Class Gone has 2 direct slot definitions:

`Http-Status-Code`

`Thing`

## 8.642 Tootsville::Gossip-Initiation

### 8.642.1 Class

Gossip-Initiation names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.642.2 Slots

Class Gossip-Initiation has 2 direct slot definitions:

Uuid

Answer

## **8.643 Tootsville::Gossip-Initiation-Answer**

### **8.643.1 Function**

Gossip-Initiation-Answer names an undocumented function, with lambda list (INSTANCE).

### **8.643.2 File**

Defined in file src/gossip.lisp.

### **8.643.3 SetF Function**

(SETF Gossip-Initiation-Answer) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.643.4 File**

Defined in file src/gossip.lisp.

## **8.644 Tootsville::Gossip-Initiation-P**

### **8.644.1 Function**

Gossip-Initiation-P names an undocumented function, with lambda list (OBJECT).

### **8.644.2 File**

Defined in file src/gossip.lisp.

## **8.645 Tootsville::Gossip-Initiation-Uuid**

### **8.645.1 Function**

Gossip-Initiation-Uuid names an undocumented function, with lambda list (INSTANCE).

### **8.645.2 File**

Defined in file src/gossip.lisp.

### **8.645.3 SetF Function**

(SETF Gossip-Initiation-Uuid) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.645.4 File**

Defined in file src/gossip.lisp.

## **8.646 Tootsville::Gracefully-Report-Error.Html**

### **8.646.1 Function**

Gracefully-Report-Error.Html names an undocumented function, with lambda list (STATUS-CODE C).

### **8.646.2 File**

Defined in file src/acceptor.lisp.

## 8.647 Tootsville::Gracefully-Report-Error.Json

### 8.647.1 Function

Gracefully-Report-Error.Json names an undocumented function, with lambda list (STATUS-CODE C).

### 8.647.2 File

Defined in file src/acceptor.lisp.

## **8.648 Tootsville::Gracefully-Report-Http-Client-Error**

### **8.648.1 Function**

Gracefully-Report-Http-Client-Error names an undocumented function, with lambda list (C).

### **8.648.2 File**

Defined in file src/acceptor.lisp.



## 8.649 Tootsville::Grant-Item

### 8.649.1 Function

Grant-Item names a function, with lambda list (TEMPLATE-ID RECIPIENT):

Create a new instance of TEMPLATE-ID and give it to RECIPIENT.

### 8.649.2 File

Defined in file src/items.lisp.

## **8.650 Tootsville::Gravatar-Hash**

### **8.650.1 Function**

Gravatar-Hash names a function, with lambda list (EMAIL):

Computes the Gravatar hash of an EMAIL address.

### **8.650.2 File**

Defined in file src/users.lisp.

## 8.651 Tootsville::Gravatar-Image-Url

### 8.651.1 Function

Gravatar-Image-Url names a function, with lambda list (EMAIL &KEY SIZE DEFAULT FORCE-DEFAULT-P RATING):

DEFAULT may be either a URL to your own image, or one of :404, :mm, :identicon, :monsterid, :wavatar, or :retro. RATING may be one of :g, :pg, :r, or :x.

### 8.651.2 File

Defined in file src/users.lisp.

## **8.652 Tootsville::Greeting/ Daemon/ Error-Output**

### **8.652.1 Function**

Greeting/ Daemon/ Error-Output names a function, with lambda list NIL:

Print a greeting to \*ERROR-OUTPUT\* (see the Common Lisp HyperSpec).

### **8.652.2 File**

Defined in file src/logging.lisp.

## **8.653 Tootsville::Greeting/ Daemon/ Log-Output**

### **8.653.1 Function**

Greeting/ Daemon/ Log-Output names a function, with lambda list NIL:

Print a greeting to the verbose info log.

### **8.653.2 File**

Defined in file src/logging.lisp.

## **8.654 Tootsville::Greeting/ Daemon/ Standard-Output**

### **8.654.1 Function**

Greeting/ Daemon/ Standard-Output names a function, with lambda list NIL:

Print a greeting to \*STANDARD-OUTPUT\* (see the Common Lisp HyperSpec).

### **8.654.2 File**

Defined in file src/logging.lisp.

## **8.655 Tootsville::Greeting/ Daemon/ Trace-Output**

### **8.655.1 Function**

Greeting/ Daemon/ Trace-Output names a function, with lambda list NIL:

Print a greeting to *\*TRACE-OUTPUT\** (see the Common Lisp HyperSpec)

### **8.655.2 File**

Defined in file `src/logging.lisp`.

## 8.656 Tootsville::Group-Plists

### 8.656.1 Function

Group-Plists names a function, with lambda list (PLISTS KEY):

Group PLISTS into a containing Alist by KEY.

Each value of KEY in the proper-list of Plists PLISTS will be an unique key in the resulting Alist.

### 8.656.2 File

Defined in file src/endpoints/slash-meta-game.lisp.



## 8.657 Tootsville::Habitat-Elevation-Roughness

### 8.657.1 Function

Habitat-Elevation-Roughness names a function, with lambda list (HABITAT):

How much relative roughness to the contour for this habitat?

### 8.657.2 File

Defined in file src/terrain.lisp.

## 8.658 Tootsville::Habitat<-Pixel

### 8.658.1 Function

Habitat<-Pixel names a function, with lambda list (R G B):

Which habitat type does the given color triplet represent?

### 8.658.2 File

Defined in file src/terrain.lisp.

## 8.659 Tootsville::Handle-Normal-Request

### 8.659.1 Function

Handle-Normal-Request names an undocumented function, with lambda list (METHOD URI-PARTS UA-ACCEPT).

### 8.659.2 File

Defined in file src/acceptor.lisp.

## **8.660 Tootsville::Handle-Options-Request**

### **8.660.1 Function**

Handle-Options-Request names an undocumented function, with lambda list (URI-PARTS UA-ACCEPT).

### **8.660.2 File**

Defined in file src/acceptor.lisp.

## **8.661 Tootsville::Harmony-Personality**

### **8.661.1 Class**

Harmony-Personality names a class, with one superclass: Section 8.1083 [TOOTSVILLE ROBOT-HARMONY], page 1368.

This class defines a character named Harmony

### **8.661.2 Slots**

Class Harmony-Personality has no direct slots defined.

## 8.662 Tootsville::Header-Time

### 8.662.1 Function

Header-Time names a function, with lambda list (&OPTIONAL (TIME (NOW))):

Get TIME in RFC-1123 format, as needed for HTTP headers.

TIME defaults to the present ((NOW)).

### 8.662.2 File

Defined in file src/types/date+time.lisp.

## **8.663 Tootsville::Holiday-Special-Personality**

### **8.663.1 Class**

Holiday-Special-Personality names a class, with one superclass: Section 8.1072 [TOOTSVILLE ROBOT], page 1357.

### **8.663.2 Slots**

Class Holiday-Special-Personality has no direct slots defined.

## 8.664 Tootsville::Host-Name-Char-P

### 8.664.1 Function

Host-Name-Char-P names a function, with lambda list (CHAR):

Is CHAR a constituent character that could be in a DNS host name?

These characters are A-Z, 0-9, or ‘.’ or - (see the Common Lisp HyperSpec) (dot or dash).

### 8.664.2 File

Defined in file `src/types/uri-types.lisp`.



## 8.665 Tootsville::Host-Name-Like-P

### 8.665.1 Function

Host-Name-Like-P names a function, with lambda list (NAME):

Does NAME meet the general rules of being a DNS host name.

Note that this does NOT recognize either dotted-quad IPv4 nor hex IPv6 addresses, only DNS names.

RFC-1035:

- Each label is up to 63 character-bytes.
- The total name length is up to 255 character-bytes, excluding dots.
- Labels must begin with a basic ASCII letter A-Z
- Labels must end with a letter or digit 0-9
- Labels may contain ASCII Hyphen-Minus, but only internally and never twice in a row.
- At present, all Top-Level Domains are at least two alphabetic characters and contain no digits nor hyphens.
- This function requires at least one dot; i.e. it is not for TLDs
- The trailing dot for the root should be omitted for this function.

### 8.665.2 File

Defined in file `src/types/uri-types.lisp`.

## **8.666 Tootsville::How-Slow-Is-Slow**

### **8.666.1 Function**

How-Slow-Is-Slow names an undocumented function, with lambda list (OBJECT).

## 8.667 Tootsville::Http-Client-Error

### 8.667.1 Class

Http-Client-Error names a class, with one superclass: COMMON-LISP::ERROR (not in this manual).

An error that can be returned to an HTTP client.

Note that we use these error codes internally, as well, so they are not necessarily always propagated over HTTP — but they could be.

### 8.667.2 Slots

Class Http-Client-Error has 1 direct slot definition:

`Http-Status-Code`

## **8.668 Tootsville::Http-Idempotent-Request-Method**

### **8.668.1 Type**

Http-Idempotent-Request-Method names a TYPE:

HTTP request methods which, if replayed, do no harm, but may yield an harmless error message on the second and subsequent attempts.

## **8.669 Tootsville::Http-Is-Success-P**

### **8.669.1 Function**

Http-Is-Success-P names an undocumented function, with lambda list (HTTP-STATUS).

### **8.669.2 File**

Defined in file `src/auth/auth-firebase.lisp`.

## **8.670 Tootsville::Http-Request-Method**

### **8.670.1 Type**

Http-Request-Method names a TYPE:

All HTTP request methods (aka verbs) defined in an IETF RFC.

## **8.671 Tootsville::Http-Safe-Request-Method**

### **8.671.1 Type**

Http-Safe-Request-Method names a TYPE:

HTTP request methods that make no changes, so can be replayed ad infinitum.

## **8.672 Tootsville::Http-Status-Code**

### **8.672.1 Function**

Http-Status-Code names an undocumented function, with lambda list (CONDITION).



## **8.673 Tootsville::Ice-Credentials**

### **8.673.1 Function**

Ice-Credentials names an undocumented function, with lambda list NIL.

### **8.673.2 File**

Defined in file src/gossip.lisp.

## **8.674 Tootsville::Ice-Url-To-Urls**

### **8.674.1 Function**

Ice-Url-To-Urls names an undocumented function, with lambda list (CREDENTIAL).

### **8.674.2 File**

Defined in file src/gossip.lisp.

## 8.675 Tootsville::Id-Column-For

### 8.675.1 Function

Id-Column-For names a function, with lambda list (TYPE):

The column (if any) providing the primary key for TYPE.

May return NIL if there is no simple primary key.

### 8.675.2 File

Defined in file src/db/generic-db.lisp.

## **8.676 Tootsville::Ignore-Duplicates**

### **8.676.1 Macro**

Ignore-Duplicates names an undocumented macro, with lambda list (&BODY BODY).

### **8.676.2 File**

Defined in file src/users.lisp.

## **8.677 Tootsville::Ignore-Not-Found**

### **8.677.1 Macro**

Ignore-Not-Found names a macro, with lambda list (&BODY BODY):

Ignore NOT-FOUND errors in BODY, and return a NIL instead.

### **8.677.2 File**

Defined in file src/db/db-central.lisp.

## 8.678 Tootsville::Infinity-Add-Furniture

### 8.678.1 Function

Infinity-Add-Furniture names a function, with lambda list (D USER RECIPIENT/S):

Alias for INFINITY-SET-FURNITURE.

Lisp ADD-FURNITURE = JSON addFurniture

Alias for Section 8.731 [TOOTSVILLE INFINITY-SET-FURNITURE], page 1009, q.v.

### 8.678.2 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.679 Tootsville::Infinity-Add-Journal-Entry

### 8.679.1 Function

Infinity-Add-Journal-Entry names a function, with lambda list (D U R):

Add a staff journal entry.

Lisp ADD-JOURNAL-ENTRY = JSON addJournalEntry

The staff journal entries are recorded to the database for later review.

UNIMPLEMENTED in 2.0.

### 8.679.2 Usage

```
{ entry: "journal text" }
```

### 8.679.3 Example

```
{ entry: "nothing to report" }
```

### 8.679.4 Romance 1.2 documentation

Staff members can create a journal entry which is stored for review in a customer service application such as Joshua. Creating a ModeratorJournal object will parse for certain values such as [[@username](#)].

### 8.679.5 Formerly Proprietary Extension

This command was formerly a proprietary extension for Tootsville.com and has now been re-created for the AGPL version of Romance.

### 8.679.6 File

Defined in file `src/infinity/tootsville-commands.lisp`.

## 8.680 Tootsville::Infinity-Add-To-List

### 8.680.1 Function

Infinity-Add-To-List names a function, with lambda list (D USER RECIPIENT/S):

Add a user to a buddy list or ignore list (removed in 1.2)

Lisp ADD-TO-LIST = JSON addToList

...using the traditional (online-only, no notification engine) mechanism (using out of band methods). Compare vs. requestBuddy Section 8.726 [TOOTSVILLE INFINITY-REQUEST-BUDDY], page 1003,

### 8.680.2 Usage

This command can no longer be used.

### 8.680.3 410 Gone

This was a legacy feature removed in Romance 1.2.

### 8.680.4 Changes from 1.1 to 1.2

This function was replaced with Section 8.726 [TOOTSVILLE INFINITY-REQUEST-BUDDY], page 1003, — requestBuddy — q.v.

### 8.680.5 File

Defined in file src/infinity/legacy-commands.lisp.



## 8.681 Tootsville::Infinity-Click

### 8.681.1 Function

Infinity-Click names a function, with lambda list (D USER RECIPIENT/S):

Used by the client to report a mouse click or finger tap.

Lisp CLICK = JSON click

### 8.681.2 Usage

If the user clicks on a placed-item, this method should be called with the following syntax:

```
{ on: ITEM-ID, x: X, y: Y, z: Z, with: MODS }
```

Note that the (x,y,z) values passed are relative to the origin point of the item; thus, if an item is placed at (200,200,200) and is clicked at (210,210,210), the coördinates reported should be (10,10,10).

If the user clicks on the ground, normally it will result in walking, but it could instead be reported as:

```
{ x: X, y: Y, z: Z, with: MODS }
```

### 8.681.3 Modifiers characters

The modifiers string can contain any of the following symbols in any order, representing modifier keys that were held down when the user clicked on the item:

‘^’

Caret represents the **Control** or **Ctrl** key on Linux<sup>®</sup> or Windows systems, or the **Command** key on macOS.

‘S’

Ess represents the **Shift** key on any platform.

‘C’

Ci represents the **Caps Lock** state being enabled. May be ignored or omitted.

‘N’

En represents the **Num Lock** state being enabled. May be ignored or omitted.

‘M’

Em represents the **Meta** key on Linux, **Alt** on Linux or Windows, or **Option** on macOS.

‘L’

Ell represents the **Scroll Lock** state being enabled. May be ignored or omitted.

‘A’

Ay represents the **Alt-Gr** key on any platform (if supported). May be ignored or omitted.

‘\*’

Asterisk represents the **Super** key on Linux or **Windows-Logo** key on Windows.

- ‘1, 2, 3’ Numbers represent mouse buttons: 1 for left, 2 for middle, 3 for right.  
 ‘+, -’ Plus represents rolling a scroll wheel down; Minus to scroll up  
 ‘<, >’ Less-than represents rolling a scroll knob left; greater-than, right.

#### 8.681.4 Flash details

In the Flash MouseEvent object, you can create the "mods" with the following:

```
var mods:String = "";
if (ev.altKey) mods += "M";
if (ev.commandKey || ev.ctrlKey) mods += "^";
if (ev.shiftKey) mods += "S";
if (ev.type == ev.CLICK) mods += "1";
if (ev.type == ev.MIDDLE_CLICK) mods += "2";
if (ev.type == ev.RIGHT_CLICK) mods += "3";
if (ev.type == ev.MOUSE_WHEEL)
{ if (ev.delta < 0) mods += "-";
  if (ev.delta > 0) mods += "+"; }
if (Keyboard.numLock) mods += "N";
if (Keyboard.capsLock) mods += "C";
```

#### 8.681.5 Changes from 1.2 to 2.0

- The ‘z’ coordinate is no longer optional.
- A form of the ‘click’ command which omitted the ‘on’ parameter.
- The itemID is an UUID Base64 string, not a moniker string — but these were always meant to be opaque identifiers.

#### 8.681.6 202 Accepted

The click event has been noted. Any outcomes of that event will be broadcast over other channels.

#### 8.681.7 204 No Content

The click event is being ignored; ITEM-ID was not an interesting item to the server.

#### 8.681.8 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.682 Tootsville::Infinity-Consider-Child-Approval

### 8.682.1 Function

Infinity-Consider-Child-Approval names a function, with lambda list (D U R):

Consider whether to approve a child's request with ID UUID.

Lisp `CONSIDER-CHILD-APPROVAL = JSON considerChildApproval`

### 8.682.2 Usage

The client sends this packet when the player, a parent or guardian who has a child Toot account on their user profile, wishes to be prompted again as to whether to approve or deny a child request with the given UUID.

```
{ uuid: "UUID-OF-REQUEST" }
```

The client will receive a `prompt` message to that effect. See Section 8.720 [TOOTSVILLE INFINITY-PROMPT-REPLY], page 991, for a discussion of the `prompt` packet and its replies.

There is no direct reply to this packet, only the asynchronous prompt (or admin error message, see below).

### 8.682.3 Error conditions

In the event of an error, the player will receive an administrative message explaining the problem. No machine-readable error packet is returned to the client.

Possible error conditions include:

- The UUID represents a child request from another user's Toot
- The UUID does not represent any current child request; we assume that this means the relevant request once existed, but was culled after it aged out — but we have no way of knowing for sure, since expired requests are hard deleted.

### 8.682.4 File

Defined in file `src/infinity/new-commands-20.lisp`.

## 8.683 Tootsville::Infinity-Create-User-House

### 8.683.1 Function

Infinity-Create-User-House names a function, with lambda list (D USER RECIPIENT/S):

Either claim the user's house and lot, or add a room to their house.

Lisp CREATE-USER-HOUSE = JSON createUserHouse

### 8.683.2 Usage

```
{ lot: "Lot ID",
  house: "House ID" }
```

```
{ index: ROOM-INDEX,
  connectTo: ROOM-INDEX,
  connectAt: "point moniker" }
```

Data describing the user's lot.

When the player has found an empty lot and wishes to claim it as their own, they choose a base house item and send

```
{ lot: lot-ID, house: house-ID }
```

When the player has a house and wishes to add a room, they send

```
{ index: roomIndex,
  connectTo: roomIndex,
  connectAt: pointMoniker }
```

WRITEME

### 8.683.3 201 Created

A house or room was created as demanded

### 8.683.4 409 Conflict

A house already exists on that lot, or, a room is already connected at the given connection point. The request cannot be completed because something already exists where the new construction was meant to be placed.

### 8.683.5 404 Not Found

The house ID or room connection point given was not found.

### 8.683.6 Changes from 1.2 to 2.0

In 1.2 adding a room required only an index.

### 8.683.7 Changes from 1.1 to 1.2

In 1.1, houses could have only one room

### 8.683.8 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.684 Tootsville::Infinity-Delete-Mail-Message

### 8.684.1 Function

Infinity-Delete-Mail-Message names a function, with lambda list (D U R):

Delete a message from the user's (SMS) mailbox

Lisp DELETE-MAIL-MESSAGE = JSON deleteMailMessage

### 8.684.2 Usage

WRITEME

### 8.684.3 Example

WRITEME

### 8.684.4 Changes from 1.2 to 2.0

id was previously an integer, but is now an UUID string.

### 8.684.5 Formerly Proprietary Extension

This command was formerly a proprietary extension for Tootsville.com and has now been re-created for the AGPL version of Romance.

### 8.684.6 File

Defined in file src/infinity/tootsville-commands.lisp.

## 8.685 Tootsville::Infinity-Doff

### 8.685.1 Function

Infinity-Doff names a function, with lambda list (D USER RECIPIENT/S):

Remove clothes or Pivitz.

Lisp DOFF = JSON doff

### 8.685.2 Usage

```
{ slot: "SLOT" }
```

```
{ type: "TYPE" }
```

### 8.685.3 Example

```
{ type: "pivitz" }
```

```
{ slot: "1D94E6C7-8643-48AE-81A4-8B0C3EB36A7A" }
```

When `type` is present, it must be either `clothes` or `pivitz`.

When `slot` is present, indicates a specific item UUID to remove.

### 8.685.4 400 Bad Request

Exactly one of `type` or `slot` must be present.

If present, `type` must be `clothes` or `pivitz`.

### 8.685.5 200 OK

Responds with total wardrobe as per Section 8.742 [TOOTSVILLE INFINITY-WARDROBE], page 1026,

### 8.685.6 Formerly Proprietary Extension

This command was formerly a proprietary extension for Tootsville.com and has now been re-created for the AGPL version of Romance.

### 8.685.7 File

Defined in file `src/infinity/tootsville-commands.lisp`.

## 8.686 Tootsville::Infinity-Dofff

### 8.686.1 Function

Infinity-Dofff names a function, with lambda list (D USER RECIPIENT/S):

Doff all clothing items.

Lisp DOFFF = JSON dofff

See also Section 8.685 [TOOTSVILLE INFINITY-DOFF], page 944, for single items. To put on (don) an item, see Section 8.687 [TOOTSVILLE INFINITY-DON], page 946. Mnemonic: Like `doff` but more so.

### 8.686.2 Usage

This command takes no parameters.

### 8.686.3 Limitations

This does not un-equip an item held in the TRUNK. This does not remove or alter a Toot's pattern. For non-Toot avatars, this does not un-equip an item held in the HAND, LHAND, or RHAND.

Sends two responses: a success reply from `dofff`, then total avatar info from `wardrobe`. See Section 8.742 [TOOTSVILLE INFINITY-WARDROBE], page 1026.

### 8.686.4 Status 200 OK

All clothing items have been removed.

```
{ from: "dofff",  
  status: true }
```

A separate `wardrobe` packet will be sent.

### 8.686.5 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.687 Tootsville::Infinity-Don

### 8.687.1 Function

Infinity-Don names a function, with lambda list (D USER RECIPIENT/S):

Don (or equip) an item

Lisp DON = JSON don

### 8.687.2 Usage

```
{ slot: "item-UUID",
  [ color: "color ID" ] }
```

JSON object has the item UUID number to be worn (clothes, pivitz, trunk).

See Section 8.1338 [TOOTSVILLE WEAR-SLOT-INFO], page 1625, for descriptions of how wear slots are identified and described. Note that the appropriate wear slot can be determined from the item's template; see Section 8.787 [TOOTSVILLE ITEM-TEMPLATE-INFO], page 1072. For a list of all wear slots, see Section 8.690 [TOOTSVILLE INFINITY-ENUMERATE-WEAR-SLOTS], page 952, (new in 2.0).

Response with total avatar info from `wardrobe`. See Section 8.742 [TOOTSVILLE INFINITY-WARDROBE], page 1026.

Color ID is no longer allowed; it will be rejected.

### 8.687.3 200 OK

The item has been donned or equipped.

### 8.687.4 400 Bad Request

The removed color attribute was submitted.

### 8.687.5 404 Not Found

The item UUID specified (by "slot") was not recognized.

### 8.687.6 403 Forbidden

The item UUID specified was not owned by the player requesting to don it.

### 8.687.7 409 Conflict

The item requested cannot be equipped by the player's avatar. For example, a Toot character cannot equip an item which requires a `HAND` slot, since Toots have no fingers. Items which do not occupy a wear slot also cannot be equipped, e.g. a tree.

### 8.687.8 Changes from 1.2 to 2.0

Colors of items can no longer be changed when donning them. This was meant for pattern changing in 1.2, which must now be accomplished in-game via Doodle. The `color` parameter must be null or absent.

Patterns are no longer clothing items.



**8.687.9 Changes from 1.0 to 1.1**

Equipment held in the TRUNK is now explicitly supported as a distinct wear slot with specific meaning (ie, the user can activate that item).

**8.687.10 File**

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.688 Tootsville::Infinity-Echo

### 8.688.1 Function

Infinity-Echo names a function, with lambda list (D USER RECIPIENT/S):

Echoes back the supplied JSON (or ActionScript) object to the client.

Lisp ECHO = JSON echo

### 8.688.2 Usage

The datum (d) is returned identically, in a return element named literally `You said`.

This method exists solely for testing purposes.

```
{ c: "echo"
  d: DATA-TO-ECHO }
```

#### 8.688.2.1 Parameters

`jso` Any JSON object, the contents of which will be returned to the caller.

`u` The user calling (to whom the response is sent)

### 8.688.3 Example

```
⇒ { c: "echo", d: { foo: 42 } }
```

```
{ from: "echo", status: true, "You said": { foo: 42 } }
```

Note that the field name is literally `'You said'` with a space.

### 8.688.4 200 OK

The response is echoed back to the user.

### 8.688.5 Limitations

The echo packet must be less than 1,024 Unicode characters in length or it will be truncated to 1,024 characters. No warning will be issued to the user in the case of truncation.

### 8.688.6 Changes from 1.2 to 2.0

The 1kc limit was introduced in 2.0.

### 8.688.7 Known bugs

This feature is not working correctly as of version 0.6.

### 8.688.8 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.689 Tootsville::Infinity-End-Event

### 8.689.1 Function

Infinity-End-Event names a function, with lambda list (D USER RECIPIENT/S):

Attempt to end an event.

End an event begun by Section 8.738 [TOOTSVILLE INFINITY-START-EVENT], page 1020, q.v.

End an event begun by Section 8.738 [TOOTSVILLE INFINITY-START-EVENT], page 1020, q.v.

Lisp END-EVENT = JSON endEvent

### 8.689.2 Calling

```
{ c: "endEvent",
  d: { moniker: "event moniker",
      ( id | eventID ): "event ID",
      status: ( "cmp" | "cxl" ),
      [ medal: "medal", ]
      [ score: "score" ] } }
```

This command terminates an event (such as a fountain, store purchase, or minigame) which was begun with Section 8.738 [TOOTSVILLE INFINITY-START-EVENT], page 1020, qv.

The parameter `eventID` can be referenced as `id` instead — but that is deprecated (since Romance 1.0) and will (eventually) be dropped.

The status code is either `cmp`, if the event was completed in some way (successfully or otherwise), or `cxl`, if the event was canceled before it reached any ind of completion.

The `score` and `medal` parameters are optional, and depend on the type of event. They should never be submitted with a `cxl` cancel packet, and are not needed for an item purchase. They are sometimes to be used with minigames. The `score` has a special relationship with magic fountains, described below.

- In the event of a magic fountain, the client should submit a random number between 1 and 100 as the `score`. This will be ignored, and a number of peanuts will be awarded to the player.
- In the event of a purchase, neither `score` nor `medal` are required.
- Other kinds of events can pass a numeric `score`, or a string for `medal`, as appropriate to their needs.

### 8.689.3 Success Response to Canceled Event

The response to a canceled (`status: "cxl"`) event will be of the form:

```
{ from: "endEvent",
  status: true,
  ended: "eventID",
  canceled: true }
```

### 8.689.4 Success Response to Completed Event

The response to a completed (`status: "cmp"`) event will be of the form:

```
{ from: "endEvent",
  status: true,
  ended: "eventID",
  peanuts: peanuts,
  fairyDust: fairyDust,
  [ highScores: { 1: { points: points,
                    userName: "user name" },
                2: ... 24: }, ]
  totalPeanuts: total,
  totalFairyDust: total,
  [ gotHighScore: index ]
```

The `endEvent` packet for a completed event indicates:

- The event ID which was ended — typically a UUID. This matches the `eventID` returned by `startEvent` and passed back to `endEvent`.
- The relative change in peanuts and fairy dust (positive means more earned; negative means a net loss), and the player's new totals of each
- If the event is a minigame or other event that could have a high score list, up to 24 top scores are returned, each with a point score, an user name, and (if the event is that sort) possibly a medal earned. Note that it is possible to get fewer than 24 scores back; conforming clients must accept zero to at least 24.
- If the event has a high score list, and the player has earned a high score now, the index (from 1 = first place) which was achieved.

### 8.689.5 Error Responses

An error response is of the form:

```
{ from: "endEvent",
  status: false,
  eventID: "event UUID",
  err: "error code",
  error: "User-visible error message" }
```

The error code can be one of:

`cost`           The item to be purchased costs more peanuts than you have.

`badStatus`  
                  The status passed was not one of `cmp` nor `cxl`

`eventID.notFound`  
                  The event ID passed was not found

`eventID.notYours`  
                  The event ID passed represents an event started by another player

`medal.notFound`  
                  The medal passed was not valid

`score.range`

The score reported was not valid; it was not in the range of possible scores for this event.

### **8.689.6 Changes from 1.2 to 2.0**

WRITEME

### **8.689.7 File**

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.690 Tootsville::Infinity-Enumerate-Wear-Slots

### 8.690.1 Function

Infinity-Enumerate-Wear-Slots names a function, with lambda list (D U RECIPIENT/S):

Enumerates all possible wear slots for any avatar.

Lisp ENUMERATE-WEAR-SLOTS = JSON enumerateWearSlots

### 8.690.2 Usage

This command takes no arguments. It returns the wear-slots associated with the caller's avatar.

See Section 8.1338 [TOOTSVILLE WEAR-SLOT-INFO], page 1625, for the format of the reply data.

### 8.690.3 200 OK

Returns an object with `status: true`, `from: "enumerateWearSlots"`, and a key `slots` under which is an array of information about each wear slot, in the format of Section 8.1338 [TOOTSVILLE WEAR-SLOT-INFO], page 1625, q.v.

```
{ from: "enumerateWearSlots",  
  status: true,  
  slots: [ WEAR-SLOT-INFO, WEAR-SLOT-INFO, ... ] }
```

### 8.690.4 File

Defined in file `src/infinity/new-commands-20.lisp`.

## 8.691 Tootsville::Infinity-Finger

### 8.691.1 Function

Infinity-Finger names a function, with lambda list (D USER RECIPIENT/S):

Get public info for a list of Toots.

Lisp FINGER = JSON finger

For details, see the synonym 'INFINITY-GET-AVATAR-INFO'.

### 8.691.2 Usage

```
{ c: "finger", d: { key: "toot-name", ... } }
```

### 8.691.3 Reply format

```
{ from: avatars, status: true, avatars: { 0: { TOOT-INFO ... }, ... } }
```

User public information is in the format of Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534, which should be a superset of what `AbstractUser.getPublicInfo()` used to return in 1.2.

### 8.691.4 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.692 Tootsville::Infinity-Game-Action

### 8.692.1 Function

Infinity-Game-Action names a function, with lambda list (D USER RECIPIENT/S):

Send an in-world game’s action.

Lisp GAME-ACTION = JSON gameAction

These are actions that affect in-world minigames.

### 8.692.2 Usage

```
{ c: "gameAction",
  d: { game: "AEB967CB-5598-40D5-9B4A-894C9BC38501",
      action: ACTION-NAME,
      [ ... PARAMS ... ] } }
```

### 8.692.3 Example

```
{ c: "gameAction",
  d: { game: "AEB967CB-5598-40D5-9B4A-894C9BC38501",
      action: "tagYouReIt",
      tagged: "5047F44E-8B1D-4B8A-9EC6-4E1D6E1653AD" } }
```

### 8.692.4 Overview of In-World Minigames

In-world minigames generally don’t use much of a special interface, but sometimes require some kind of additional overlay. The game actions are usually signaled by in-game items.

In-world minigames include soccer, volleyball, croquet, bowling, card table games, tag, and more. Each of these games may have a score and possibly some enforceable rules, although we often leave enforcement of the rules to the players (so that they can choose which rule set they like).

These game actions are identified by function names beginning with “GAME-ACTION-.” Some of them include:

- Section 8.613 [TOOTSVILLE GAME-ACTION-START-SPORTS-BALL-GAME], page 871,
- Section 8.607 [TOOTSVILLE GAME-ACTION-JOIN-CARD-GAME], page 864,
- Section 8.615 [TOOTSVILLE GAME-ACTION-TAG-YOU-RE-IT], page 873,

### 8.692.5 General Structure

A `gameAction` packet has a `d` datum with a key `action`, which is used to further dispatch the game action to its appropriate handler. The `action` value is the `smallCamelCase` version of the “GAME-ACTION-function-name” that will actually handle it.

The specific game which is being addressed must be identified by its `UUID`. This is usually discovered by finding a game tag on an item or place in the game world.

WRITEME: Explain how to find a game tag.

Refer to the individual game action functions for further details.

See Appendix 8 for an index of game actions.



### 8.692.6 Response format

The individual game action handlers will provide their own response formats. In general, they will come from `gameAction`, with a `status` of true or false; when false, they should include an `error` text which may be user-visible, and may include an `err` tag which is a general machine-readable code.

### 8.692.7 Status 400 Error

If the `action` is not supplied, or if no such action is known to the server, then an error 400 is returned, with a JSON error packet of the usual form:

```
{ from: "gameAction",  
  status: false,  
  error: "error message text",  
  err: "game-action-not-found" }
```

### 8.692.8 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.693 Tootsville::Infinity-Get-Apple

### 8.693.1 Function

Infinity-Get-Apple names a function, with lambda list (CLIENT &OPTIONAL PACKET):

Get the apple to get into, or out of, \$Eden.

### 8.693.2 Theory

The “apple” is a seed value for a sort of CHAP/HMAC authentication used only for children signing in to Tootsville. Essentially, the client will ask for an “apple,” as a seed value for hashing with the password. The client sends back this hash, and awaits parental permission.

### 8.693.3 Apple-based authentication

In the modern usage, the user who wishes to get authenticated connects a stream (ie, WebSocket) connection and sends a packet like this:

```
{ c: "getApple" }
```

There are no d data required.

The response from the server will be something like

```
{ from: "getApple",
  status: true,
  apple: "an opaque string" }
```

The default action is to create a new apple value on each call. However, the client can control this with an additional parameter, **replace**.

#### supersede

The default. A new apple will be returned, superseding any previous value, regardless as to whether any previous value had been given. Previous apple values became irrelevant / no longer can be used.

#### never

If an apple value has been issued, do not replace it. An error will be returned on subsequent calls to **getApple**.

#### replace

Assert that there must have been a previous apple issued, and replace it. If no previous apple had been issued, an error will be returned.

The **apple** value will be a valid UTF-8 string without control characters of no more than 4kiB, but no other assertions about it can be assumed by a conforming client.

In the case of an error from **getApple**, a returned error packet will look like

```
{ from: "getApple", status: false,
  error: "error message text" }
```

Upon receiving a valid apple string, the client will submit a login packet (see: Section 8.713 [TOOTSVILLE INFINITY-LOGIN], page 983) like:

```
{ c: "login",
  d: { userName: "a-Toot-name",
      password: "a-secret-sha1-hex-string",
```

```
zone: "$Eden" } }
```

The `pass` submitted is a hash created by:

1. Concatenate the `apple` value with the downcased version of `child-code` for the Toot being signed-in.
2. Take this concatenated string, and take the SHA1 hash of it.
3. Take the hex value of that SHA1 hash

The login packet will return `from: "login"`, `status: true` if the password is successful.

Next, parental approval is required. This can be submitted before login, in which case the login will be followed by a slew of other messages as the player signs into the game, or after login. In the latter case, the client will be given the login success message and nothing else. The client is expected to wait and entertain the user until such a time as parental approval comes back.

Parental approval packets are sent by Section 8.943 [TOOTSVILLE PARENT-GRANT-PERMISSION], page 1228, by way of Section 8.1388 [TOOTSVILLE WS-APPROVE-TOOT], page 1675; denial, by Section 8.942 [TOOTSVILLE PARENT-DENY-PERMISSION], page 1227, by way of Section 8.1393 [TOOTSVILLE WS-DENY-TOOT], page 1680.

During this intermediate time between login and approval, the client's Section 8.1311 [TOOTSVILLE USER-ACCOUNT], page 1598, will be set to NIL but its Section 8.1230 [TOOTSVILLE TOOT], page 1515, will be set to the selected Toot object.

If the parent approves, a packet will be returned like

```
{ from: "parentApproval",
  status: true,
  until: UNIX-TIME,
  approved: "approved" }
```

If the parent does not approve (actively denies permission), a packet will be returned like

```
{ from: "parentApproval",
  status: false,
  until: UNIX-TIME-NOW,
  approved: "denied" }
```

Following denial, the client is required to cease attempting to log in; it is expected that the child user will be brought to the Wiki page explaining that they have been denied permission.

Following approval, a flood of related login packets will be sent which should trigger the usual login process; these will include positioning the Toot character, observations of the world (so-called “room variables” and avatar information, &c.) and other packets. Review the `Tootsville.Game.Gatekeeper` documentation for the client's handling of these packets.

### 8.693.4 New in 1.1

This mechanism for logins was introduced in 1.1

### **8.693.5 Changes from 1.1 to 1.2**

1.2 switched all communications to JSON, removing XML equivalent legacy commands used by SmartFox Server's protocol.

### **8.693.6 Changes from 1.2 to 2.0**

- Apple values may now potentially reach 4kiB; the former limit was 256 characters.
- Apple values are UTF-8, not ASCII-67 (7-bit) characters.
- Apple values will not contain control characters.
- The login zone is only `$Eden`; there are no other zones.
- Parental approval is required to proceed with each login, not a one-shot event during the sign-up process.
- Passwords are downcased to make them case-insensitive

### **8.693.7 File**

Defined in file `src/websockets.lisp`.

## 8.694 Tootsville::Infinity-Get-Avatars

### 8.694.1 Function

Infinity-Get-Avatars names a function, with lambda list (D USER RECIPIENT/S):

Get avatar data for a list of (other) users.

Lisp GET-AVATARS = JSON getAvatars

Synonym for Section 8.691 [TOOTSVILLE INFINITY-FINGER], page 953,

### 8.694.2 Usage

The `d` datum is a JSON object, with (ignored) keys tied to values which must be the names of users.

### 8.694.3 Example

```
{ c: "getAvatars",  
  d: { "foo": "mouser",  
        "bar": "catville" } }
```

### 8.694.4 Status 200 OK

The avatar information for each user requested will be returned in an associative array object with the same keys as the source query. The values of each key are the avatar data as returned by Section 8.691 [TOOTSVILLE INFINITY-FINGER], page 953, i.e. the information returned by Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534.

### 8.694.5 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.695 Tootsville::Infinity-Get-Color-Palettes

### 8.695.1 Function

Infinity-Get-Color-Palettes names a function, with lambda list (D USER RECIPIENT/S):

```
getColorPalettes
```

```
Lisp GET-COLOR-PALETTES = JSON getColorPalettes
```

### 8.695.2 Usage

This command requires no parameters

### 8.695.3 Status 410 Gone

Removed.. This routine appeared to be unused by anyone in Romance 1.1 and was removed in 1.2.

returns palettes in "extraColors", "baseColors", "patternColors" in the JSON result object (from: "getColorPalettes")

### 8.695.4 Changes from 1.1 to 1.2

Not used in Tootsville any more. The analogous palettes in Li'l Vampies and Empires of the Air are being replaced with algorithmic checks, so this routine was removed in Romance 1.2.0.

### 8.695.5 Revival?

This might be revived in 2.0 for the UI to present lists of named colors during character creation, rather than using hard-coded lists that have to be separately maintained in the client and server both.

### 8.695.6 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.696 Tootsville::Infinity-Get-Inventory

### 8.696.1 Function

Infinity-Get-Inventory names a function, with lambda list (D USER RECIPIENT/S):

Get all inventory for an user (themselves) — both active and inactive

Lisp GET-INVENTORY = JSON getInventory

### 8.696.2 Usage

This command requires no parameters.

```
{ c: "getInventory" }
```

### 8.696.3 Status 200 OK

Returns a set of items as

```
{ from: "getInventory",  
  inv: { 0: { id: 123, isActive: boolean }, ... } }  
WRITEME
```

### 8.696.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.697 Tootsville::Infinity-Get-Inventory-By-Type

### 8.697.1 Function

Infinity-Get-Inventory-By-Type names a function, with lambda list (D USER RECIPIENT/S):

Get a subset of items from your own inventory

Lisp GET-INVENTORY-BY-TYPE = JSON getInventoryByType

### 8.697.2 Usage

```
{ c: "getInventoryByType",
  d: { type: TYPE,
      [withActive: BOOLEAN ],
      [who: LOGIN-NAME ] } }
```

The `type` can be one of two options.

For legacy compatibility, the following list of `type` codes can be supplied. These may be more convenient for the front-end. Legacy users of code sequences beginning with `#` or `$` are no longer supported, however.

`clothes`

All items which can be worn in any slot other than TRUNK, HAND, LHAND or RHAND, or PIVITZ

`pivitz`

Only Pivitz items

`patterns`

Ignored for backward-compatibility.

`furniture`

Any item which cannot be equipped in any way

`structure`

Ignored for backward-compatibility

`music`

Ignored for backward-compatibility

`tootsBook`

Ignored for backward-compatibility

`stationery`

Ignored in 2.0 but may be revived in 2.1

`accessories`

All items which can be equipped in TRUNK slot, or HAND, LHAND or RHAND (for non-Toot characters).

In addition, `type` can be a string containing the word `point` followed by a space and the moniker of an avatar attachment point, in which case all items which can be equipped to that point (regardless of valence) are returned; or, the word `slot` followed by a space and the ID number of a specific wear-slot.



Finally, multiple codes can be enumerated by passing as string beginning with `$` plus a series of identifiers from the above delimited by `:`, e.g. `$clothes:pivitz`.

You can also supply `withActive: false` to screen out active items.

The optional parameter `who` specifies whose inventory to list. If not specified, the inventory of the Toot posing the question is returned. Note that inactive items of Toots not owned by you are generally not returned.

### 8.697.3 Changes from 1.2 to 2.0

In Romance 1.2, the `type` code could not be a `point` or `slot`, but it could be a string beginning `#` with a list of type code numbers; e.g. `#2:3`.

In Romance 1.2, placed furniture was also returned; this is no longer the case.

### 8.697.4 Status 200 OK

Returns a set of items as

```
{from: "inventory", for: USER-LOGIN, type: TYPE-QUERY
  inv: { 0: { id: 123, isActive: boolean },
  ... }
```

See GET-INVENTORY-BY-TYPE (UNIMPLEMENTED)

### 8.697.5 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.698 Tootsville::Infinity-Get-Mail-In-Box

### 8.698.1 Function

Infinity-Get-Mail-In-Box names a function, with lambda list (D U R):

Get a listing of messages in an SMS mailbox.

Lisp GET-MAIL-IN-BOX = JSON getMailInBox

### 8.698.2 Usage

```
{ [ from: INDEX ],
  [ limit: COUNT ] }
```

### 8.698.3 Examples

```
{ }
```

```
{ from: 10, limit: 10 }
```

```
{ from: 0, limit: 100 }
```

```
{ limit 100 }
```

```
{ from: 10 }
```

When specified, `from` is the index of the first message to return, and `limit` is the number of messages to return.

`limit` defaults to 100 messages if not supplied.

### 8.698.4 200 OK

Returns an object named `mail`. Keys under `mail` are indices. Each message consists of

<code>id</code>	An UUID for the message
<code>from</code>	The sender's name
<code>to</code>	The recipient's name
<code>subject</code>	No longer used; always ""
<code>sentTime</code>	The date and time sent
<code>readTime</code>	The date and time first (previously) retrieved by the client. Messages will be marked as read "now" when retrieved, but only after they are retrieved for the first time.
<code>body</code>	The contents of the message.

### 8.698.5 416 Request Range Not Satisfiable

The `from` value exceeded the maximum message in the Toot's inbox.

### 8.698.6 Changes from 1.2 to 2.0

Message ID's are now UUID's. Messages no longer have subjects.

**8.698.7 Formerly Proprietary Extension**

This command was formerly a proprietary extension for Tootsville.com and has now been re-created for the AGPL version of Romance.

**8.698.8 File**

Defined in file `src/infinity/tootsville-commands.lisp`.

## 8.699 Tootsville::Infinity-Get-Online-Users

### 8.699.1 Function

Infinity-Get-Online-Users names a function, with lambda list (D USER RECIPIENT/S):

Get a list of users online.

```
Lisp GET-ONLINE-USERS = JSON getOnlineUsers
```

This is an administrative function, only available to staff members.

### 8.699.2 Usage

```
{ c: "getOnlineUsers",
  d: { [ inRoom: ROOM ] }
```

If this contains an attribute of "inRoom" with a room moniker, we'll only return the users in that room. Otherwise, all users in the Zone will be returned.

This optional parameter should not be specified and will be ignored if present.

### 8.699.3 Example

```
{ c: "getOnlineUsers" }
```

### 8.699.4 Status 200 OK

```
WRITEME
```

### 8.699.5 Status 403 Permission Denied

This is returned if the user is not a Builder Toot.

```
{ from: "getOnlineUsers",
  status: false,
  error: "That is a Builder Toot command." }
```

### 8.699.6 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.700 Tootsville::Infinity-Get-Passport

### 8.700.1 Function

Infinity-Get-Passport names a function, with lambda list (D U R):

Get the list of places that the user has gotten a passport stamp at.

Lisp GET-PASSPORT = JSON getPassport

Passport stamps are not currently implemented but will be returning.

See Section 8.1261 [TOOTSVILLE TOOT-PASSPORT-STAMPS], page 1548.

### 8.700.2 Usage

This command requires no parameters.

### 8.700.3 200 OK

WRITEME

The reply format is a WRITEME but should be unchanged from 1.2.

### 8.700.4 Changes from 1.2 to 2.0

Passports stamps are temporarily unavailable.

### 8.700.5 Formerly Proprietary Extension

This command was formerly a proprietary extension for Tootsville.com and has now been re-created for the AGPL version of Romance.

### 8.700.6 File

Defined in file src/infinity/tootsville-commands.lisp.

## 8.701 Tootsville::Infinity-Get-Room-List

### 8.701.1 Function

Infinity-Get-Room-List names a function, with lambda list (D USER RECIPIENT/S):

Get a list of all “well known” Rooms currently active/visible.

Lisp GET-ROOM-LIST = JSON getRoomList

“Rooms” no longer exist. The “rooms” are now known as “spots.”

UNIMPLEMENTED

### 8.701.2 Usage

```
{ c: "getRoomList" }
```

### 8.701.3 Status 200 OK

WRITEME

### 8.701.4 Changes from 1.2 to 2.0

The system used to be broken into “rooms,” each about one “screen” size, and communications and game events were mostly restricted to the room in which they occurred.

This is no longer the case.

However, “named spots” have been introduced, so this function was repurposed to that end.

### 8.701.5 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.702 Tootsville::Infinity-Get-Room-Vars

### 8.702.1 Function

Infinity-Get-Room-Vars names a function, with lambda list (D U RECIPIENT/S):

Returns “room variables.”

Lisp GET-ROOM-VARS = JSON getRoomVars

### 8.702.2 Usage

This command requires no parameters.

### 8.702.3 Historical Usage (Romance I)

In Romance I, the server had a library of free-form key-value pairs which were used to control each “room,” or screen, of the game.

These variables, which were usually edited using the special “Zookeeper” client by Eric Feilding, eventually metamorphosed into a library of very specific “room variables” as described herein.

We no longer support arbitrary key-value pairs; at this point, all room variables are specifically enumerated in the following documentation; however, future releases could expand this list, so conforming clients are required to accept and ignore unrecognized variables silently.

### 8.702.4 Room Environment

These room variables define the general environment.

**s**

The Sky. Consists of the background (sky) texture file as a URL, or, the position of a sky object such as the sun, a moon, or a cloud.

**f**

The Floor; no longer used in 2.0. (This was the actual SWF file that had the room background in it, in Romance I.)

**w**

The Weather, or overlay artwork. Used to indicate precipitation.

### 8.702.5 Sky Variables

WRITEME

See Section 8.1137 [TOOTSVILLE SKY-ROOM-VAR], page 1422,

### 8.702.6 Weather

WRITEME

### 8.702.7 Room Objects

**item**

A placed item can be represented by an encoded string form (“item”), or a JSON structure (“itm2”).

The older style uses a key beginning with `item` and an unique identifier string, followed by a `~` delimited list of: description, X position, Y position, facing, and (optional) Z position.

If the Z position is omitted, then the value given for Y position should be used for Z instead. (The Y axis used to run across the floor.)

The facing value can be given in radians, or as a special moniker from the set: N NE E SE S SW W NW. See Section 8.750 [TOOTSVILLE INTERPRET-FACING], page 1035.

```
itemfoo123: "flowerPot~100~931~N"
itembar456: "flowerPot~100~0~1.23412952423~931"
```

#### itm2

Placed items, new form: JSON object

```
{ uuid:
  position: { x: y: z: },
  facing: FACING,
  baseColor: COLOR,
  altColor: COLOR,
  energy: NUMBER,
  scale: { x: y: z: },
  world: { world: lat: long: alt: },
  template:
  { id:
    name:
    description:
    trade: [ "Y", "N", or "X" ],
    avatar:
    energyKind:
    energyMax:
    onZero:
    wearSlot:
    weight: } }
```

#### furn

User-positioned items: key: “furn” — no longer used.

#### text

Text items: key: "text" + unique-ID = value

Text to be displayed atop another item. The value might be `x~z~string` or `itm2-id~attachment~string`. In the latter form, the text is attached to the model of the “itm2” given at the attachment point.

The attachment point is expected to be of the form `tex:TEXTURE-NAME`, i.e. a literal prefix `tex:` followed by the name of the surface texture onto which the text should be drawn.

### 8.702.7.1 Changes from 1.2 to 2.0

The facing directions can now be cardinal directions, or radians.



User-placed “furniture” is no longer distinguished from other items in the world.

`text` items can now be associated with items, rather than having fixed positions of their own.

### 8.702.7.2 Changes from 1.1 to 1.2

The `itm2` format was added.

### 8.702.8 Places

Places are regions of the game space defined by polygonal outlines. These are held in Room Variables with names of the form "zone" plus an arbitrary identifier. The contents of the room variable are a *key* followed by ":" and a series of coördinates.

Each coördinate pair/triplet is given as x,y,z in decimal, literally, like: "100,0,200". When only two coördinates are supplied, they represent x and z. They are separated with "~". To stop one polygon and start on another, give "~~" with no coördinates between.

The key of a Place specifies its purpose. The keys understood by the server include:

`grass`

This is the default Place kind; any area of ground that is not explicitly part of some other kind of Place is grass.

`tallGrass`

`water`

`unwalkable`

This demarcates an invisible obstacle — a collision-only object — which prevents avatars from entering that space.

`doormat`

`parking`

`driveway`

`stairs`

`sidewalk`

`cobbles`

`slide`

`firepole`

`game`

This space is part of an in-world game; e.g. a soccer field.

`ice`

`sand`

`snow`

`cheese`

The stuff the moons are made of. (Fight me.)

`pit`

A bottomless pit

### 8.702.8.1 Changes from 1.2 to 2.0

Places (referred to, confusingly, as zones) existed in Romance 1.0, but they came in two forms. Some zones were “burned in” to the Flash “floor” files as invisible polygon layers with a specific naming convention. Others were promulgated by room variables.

The variety of places has been substantially increased.

The default was for the “floor” to be *unwalkable*, with walkable spaces marked out by zones. The reverse is now true, however, items are now physical boundaries that block player movement.

### 8.702.8.2 Changes from 1.0 to 1.1

Prior to 1.1, all floor zones were embedded permanently in the Flash “floor” files.

### 8.702.9 More good stuff

WRITEME — there is more to explain about room variables.

### 8.702.10 See Also

See Section 7.61 [TOOTSVILLE-USER PLACE], page 189, for an explanation of creating certain places in the game and how they work.

### 8.702.11 File

Defined in file `src/infinity/new-commands-20.lisp`.

## 8.703 Tootsville::Infinity-Get-Server-Time

### 8.703.1 Function

Infinity-Get-Server-Time names a function, with lambda list (D USER RECIPIENT/S):

Send the server time to the client requesting it

Lisp GET-SERVER-TIME = JSON getServerTime

For synchronization purposes.

Sends a JSON object with a property, `serverTime`, with the current time in milliseconds (give or take transit time). This is the Unix time, not the Universal time, and in milliseconds, not seconds.

### 8.703.2 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.704 Tootsville::Infinity-Get-Session-Apple

### 8.704.1 Function

Infinity-Get-Session-Apple names a function, with lambda list (D USER RECIPIENT/S):

Initialise a session key for stream or batch mode operations.

Lisp GET-SESSION-APPLE = JSON getSessionApple

Note that this command is still available, but only in the pre-login phase of communications; once signed it, it will signal an error if called.

### 8.704.2 410 Gone

This function is no longer needed.

### 8.704.3 New in 1.1

This feature was added in Romance 1.1 and removed in 2.0

### 8.704.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.705 Tootsville::Infinity-Get-Store-Item-Info

### 8.705.1 Function

Infinity-Get-Store-Item-Info names a function, with lambda list (D USER RECIPIENT/S):

Get information about items in a store which can be purchased.

Lisp GET-STORE-ITEM-INFO = JSON `getStoreItemInfo`

Input: `jso` - JavaScript array-style object where the key names are insignificant, but the values are store item ID's

The returned packet is from: `"getStoreItemInfo"` and contains an object `items` with a matching set of keys, but whose values are objects in the form of 'STORE-ITEM-INFO', `qv`.

### 8.705.2 Changes from 1.2 to 2.0

Additional information is returned in 'STORE-ITEM-INFO' objects.

### 8.705.3 200 OK

Returns the details about store items queried-for by the user.

### 8.705.4 404 Not Found

If any item ID cannot be found, the entire query fails with a 404.

### 8.705.5 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.706 Tootsville::Infinity-Get-User-Lists

### 8.706.1 Function

Infinity-Get-User-Lists names a function, with lambda list (D USER RECIPIENT/S):

Get the user's buddy list and ignore list.

Lisp GET-USER-LISTS = JSON getUserLists

```
{ buddyList: { ... } , ignoreList: { ... } }
```

### 8.706.2 Changes from 1.2 to 2.0

Buddies on the buddy list can be starred, with attribute `starred: true`.

### 8.706.3 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.707 Tootsville::Infinity-Get-Wallet

### 8.707.1 Function

Infinity-Get-Wallet names a function, with lambda list (D USER RECIPIENT/S):

Get the contents of the player's wallet (peanuts and fairy dust)

Lisp GET-WALLET = JSON getWallet

Returns information in the form Section 8.1332 [TOOTSVILLE WALLET-INFO], page 1619, qv.

### 8.707.2 Changes from 1.1 to 1.2

Currencies were made explicit, allowing currencies other than peanuts to be potentially supported in future.

### 8.707.3 Changes from 1.2 to 2.0

Fairy Dust was added after 1.2.

### 8.707.4 200 OK

Returns the wallet info.

### 8.707.5 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.708 Tootsville::Infinity-Get-Zone-List

### 8.708.1 Function

Infinity-Get-Zone-List names a function, with lambda list (D USER RECIPIENT/S):

Get a list of all Zones currently active/visible.

Lisp GET-ZONE-LIST = JSON getZoneList

This returns "Universe" as the only Zone.

### 8.708.2 Changes from 1.2 to 2.0

Zones no longer exist.

### 8.708.3 File

Defined in file src/infinity/legacy-commands.lisp.



## 8.709 Tootsville::Infinity-Give

### 8.709.1 Function

Infinity-Give names a function, with lambda list (D USER RECIPIENT/S):

Give an item to another user.

Lisp GIVE = JSON give

XXX: notify the recipient using notifications (currently using a Message Box popup message)

```
jso - { slot: ITEM-UUID, to: TOOT-NAME }
```

```
u - giver
```

If the item is currently equipped or being worn, it will be unequipped as it is being given away.

### 8.709.2 412 Precondition Failed

An item cannot be given if you do not possess it to begin with.

### 8.709.3 404 Not Found

The item and the recipient must each exist.

### 8.709.4 403 Forbidden

Certain items cannot be traded. This includes gifting, dropping, &c. See 'ITEM-TEMPLATE-CAN-TRADE-P' for a discussion.

### 8.709.5 Changes from 1.2 to 2.0

Players used to be unable to gift items to non-VIT members; with the abolition of VIT status, everyone is very important and can receive items.

### 8.709.6 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.710 Tootsville::Infinity-Go

### 8.710.1 Function

Infinity-Go names a function, with lambda list (D USER RECIPIENT/S):

go to a place and/or perform a gesture

Lisp GO = JSON go

```
{ do: VERB (required)
  x: DEST, y: DEST, z: DEST (each optional, but if one is given, all 3 must be)
  facing: FACING (optional)
}
```

The facing can be given as per Section 8.750 [TOOTSVILLE INTERPRET-FACING], page 1035.

### 8.710.2 Changes from 1.2 to 2.0

z can no longer be omitted if x or y are specified. In 1.2, a pair with only x,y was valid.

### 8.710.3 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.711 Tootsville::Infinity-Init-User-Room

### 8.711.1 Function

Infinity-Init-User-Room names a function, with lambda list (D USER RECIPIENT/S):

Create a user's private room (in their house).

Lisp INIT-USER-ROOM = JSON initUserRoom

Creates room named user/user's name/room — room is the room index number given in the JSON data as "room," it will always be zero right now as all users have single-room houses. This will populate all furniture-type items for that room onto a set of room variables owned by the user. The user calling this method must be the owner of the room. If the user has not visited his/her house before, this will return an asynchronous "make a new house" notification to do the "first run" screen, by sending a message of type

```
{ "from": "initUserRoom",
  "status": false, "err": "showFirstRun" }.
```

Success: responds with true, and "moniker": the room's moniker (user/WHOEVER/123)■

If unnecessary, returns an error of "exists" meaning that the room is already existing

```
jso - { room: (room-number), autoJoin: (boolean) }
```

u - The user whose house-room needs to be initialized

### 8.711.2 410 Gone

Removed in 2.0.

User rooms are no longer needed nor supported.

### 8.711.3 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.712 Tootsville::Infinity-Join

### 8.712.1 Function

Infinity-Join names a function, with lambda list (D USER RECIPIENT/S):

Join a room.

Lisp JOIN = JSON join

No longer needed. We no longer have rooms.

### 8.712.2 Usage

```
{ c: "join", d: { room: NEW-ROOM, [ from: OLD-ROOM ] } }
```

### 8.712.3 Status 200 OK

```
{ from: "roomJoin",
  status: true,
  room: MONIKER }
```

You will never get this reply in Romance 2.0.

NOTE the inconsistency: the command is join, but the reply comes from roomJoin

### 8.712.4 Error Return values

zone.notFound

The user is not in a Zone

room.noMoniker

No room moniker was given to be joined

room.notFound

The room moniker does not refer to an actual room in this Zone

room.full

The room is too full (too many users)

### 8.712.5 410 Gone

Removed in 2.0.

Attempting to call join will always result in

```
{ from: "roomJoin",
  status: false,
  err: "room.notFound",
  error: "There are no rooms in Tootsville V." }
```

### 8.712.6 Changes from 1.2 to 2.0

In Romance 1.2, the room was divided into “rooms.” This is no longer the case, so there is never any need to join a room.

The success and error return codes are documented here for completeness, but only `room.notFound` will be returned.

### 8.712.7 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.713 Tootsville::Infinity-Login

### 8.713.1 Function

Infinity-Login names a function, with lambda list (CLIENT PACKET):

Notification of a new player in the game.

See Section 8.693 [TOOTSVILLE INFINITY-GET-APPLE], page 956, for an overview of the login process.

Response: logOK or { from: "login", status: false, err: "login.fail", msg: reason }

### 8.713.2 Usage

```
{ userName: LOGIN,
  password: SHA1-HEX,
  zone: "$Eden" }
```

The input packet must have 3 data elements:

**userName**

The name of the Toot character signing in

**password**

The SHA1 hex hash of the concatenated apple and password values (see Section 8.693 [TOOTSVILLE INFINITY-GET-APPLE], page 956, for details)

**zone**

Must always be \$Eden exactly.

In the event of failure, see Section 8.830 [TOOTSVILLE LOGIN-FAIL], page 1115, for possible failure (error) codes that can be returned.

### 8.713.3 Example

```
{ c: "login",
  d: { userName: "Pil",
      password: "6b4cd72086d278a9a0df40de7b4011fcea538dd",
      zone: "$Eden" } }
```

### 8.713.4 Changes from 1.2 to 2.0

In 1.2, users would log in to zone \$Eden, then log in again to a specific zone. Now, \$Eden is just a placeholder and there are no sharded zones.

Login does not completely succeed without parental approval.

The `err2` value was added to error packets for better client software support.

### 8.713.5 Changes from 1.1 to 1.2

Password hashing used the MD5 digest, which is no longer considered strong enough for Tootsville security.

### 8.713.6 Changes from 1.0 to 1.1

The Section 8.693 [TOOTSVILLE INFINITY-GET-APPLE], page 956, system was implemented. Previously, users submitted unsalted hashes of their password, which was (potentially) subject to replay attacks.

**8.713.7 File**

Defined in file src/websockets.lisp.

## 8.714 Tootsville::Infinity-Logout

### 8.714.1 Function

Infinity-Logout names a function, with lambda list (D USER RECIPIENT/S):

Log out of this game session

Lisp LOGOUT = JSON logout

### 8.714.2 Changes from 1.2 to 2.0

There was a bug in the Persephone client that caused it to explode if we logged it out before it received & processed the logout message. So, we waited for the expected lag time to expire and then throw 2 full seconds of wasted wait time after it, which had ought to be enough time. This is no longer supported.

### 8.714.3 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.715 Tootsville::Infinity-Mail-Customer-Service

### 8.715.1 Function

Infinity-Mail-Customer-Service names a function, with lambda list (D USER RECIPIENT/S):

Send an eMail to customer service (feedback)

Lisp MAIL-CUSTOMER-SERVICE = JSON mailCustomerService

UNIMPLEMENTED

This sends an email with the given subject and body to `support@Tootsville.org`.

### 8.715.2 Usage

```
{ subject: STRING, body: STRING }
```

### 8.715.3 File

Defined in file `src/infinity/legacy-commands.lisp`.



## 8.716 Tootsville::Infinity-Peek-At-Inventory

### 8.716.1 Function

Infinity-Peek-At-Inventory names a function, with lambda list (D USER RECIPIENT/S):

Look at other users' inventories

Lisp PEEK-AT-INVENTORY = JSON peekAtInventory

When requesting the inventory of another player, only their public inventory will be returned.

The optional type code is as per Section 8.697 [TOOTSVILLE INFINITY-GET-INVENTORY-BY-TYPE], page 962.

### 8.716.2 Usage

```
{ who: LOGIN-NAME, [ type: TYPE-CODE ] }
```

### 8.716.3 Examples

```
{ who: "user-name" }
```

```
{ who: "user-name",
  type: "type-code" }
```

### 8.716.4 Status 200 OK

```
{ from: "peekAtInventory",
  status: true,
  for: USER-NAME,
  inv: { 0: ITEM-INFO, [ ... ] } }
```

### 8.716.5 Status 404 Not Found

The user name given was not found.

```
{ from: "peekAtInventory",
  status: false,
  err: "login.notFound",
  error: "There is no user named LOGIN" }
WRITEME
```

### 8.716.6 Status 400 Argument Error

The type code given was not understood

```
{ from: "peekAtInventory",
  status: false,
  err: "typeCode.notFound",
  error: "Parameter error: The type code given is not recognized." }
```

### 8.716.7 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.717 Tootsville::Infinity-Ping

### 8.717.1 Function

Infinity-Ping names a function, with lambda list (D USER RECIPIENT/S):

Send a ping to the server to get back a pong.

Lisp PING = JSON ping

This also updates the user's last-active timestamp.

### 8.717.2 Usage

```
{ [ pingStarted: TIMESTAMP ] }
```

### 8.717.3 Examples

```
{ pingStarted: 1589849202000 }
```

```
{ }
```

### 8.717.4 200 OK

The response packet contains literally

```
from      "ping"
status    true
ping      "pong"
pingStarted
          see below

serverTime
```

The server's time as a Unix-epoch timestamp in milliseconds.

If the user sends a `pingStarted` value, it is replied back unchanged; otherwise, `pingStarted` is replied with the server-time as well.

### 8.717.5 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.718 Tootsville::Infinity-Play-With

### 8.718.1 Function

Infinity-Play-With names a function, with lambda list (D U R):

Choose a Toot as your active CHARACTER in the game.

Lisp PLAY-WITH = JSON playWith

CHARACTER must be the name of a Toot character owned by \*USER\*.

### 8.718.2 Usage

```
{ c: "playWith", d: { character: "a-Toot-name" } }
```

### 8.718.3 Status 200 OK

```
{ from: "playWith",  
  status: true }
```

This calls Section 8.997 [TOOTSVILLE PLAY-WITH-TOOT], page 1282, upon success, q.v.

### 8.718.4 Status 403 Not Your Toot

\*USER\* must be the owner of the Toot named CHARACTER, or you will be denied permission.

```
{ from: "playWith", status: false, error: "Not your Toot" }
```

### 8.718.5 Status 404 No Such Toot

The Toot named CHARACTER must exist.

```
{ from: "playWith", status: false, error: "No such Toot" }
```

### 8.718.6 File

Defined in file src/infinity/new-commands-20.lisp.

## 8.719 Tootsville::Infinity-Pre-Login

### 8.719.1 Function

Infinity-Pre-Login names a function, with lambda list (C AUTH CLIENT):

Handle  $\infty$  mode pre-login commands.

Commands supported:

`getApple`

See Section 8.693 [TOOTSVILLE INFINITY-GET-APPLE], page 956,

`login`

See Section 8.713 [TOOTSVILLE INFINITY-LOGIN], page 983,

Commands ignored with error returns for compatibility with version 1.2:

`batch`

`finger`

`getZoneList`

### 8.719.2 Changes from 1.0 to 1.2

I don't actually have a record as to when these commands were added, but `batch` and `finger` were added in either 1.1 or 1.2.

`batch` was used for scripting server events from shell scripts, and `finger` was used by the Toot Viewer application to obtain Toot public information without logging in. Any new Toot Viewer can use the REST interface for that purpose.

### 8.719.3 Changes from 1.2 to 2.0

Several commands are no longer supported. Note also changes to the `getApple` protocol at Section 8.693 [TOOTSVILLE INFINITY-GET-APPLE], page 956, and the new Alef-null external authentication at 'GET-USER-FOR-JSON'. Due to reliance upon external authentication services, Romance now *only* accepts password-based CHAP authentication (the `$Eden/getApple` protocol) from child Toots. This is not much of a change, since most Toots in Tootsville IV were children, but on the other hand it represents a major change in expecting more adult players and registration of parents.

`batch` is no longer supported. The REST interfaces serve the same purpose with far less complexity.

`finger` is no longer supported in this way. See Section 8.691 [TOOTSVILLE INFINITY-FINGER], page 953, for a REST endpoint with authentication, or Section 8.418 [TOOTSVILLE ENDPOINT-GET-/ toots/ toot-name→json], page 674, for a REST interface to obtain public Toot information without authentication.

`getZoneList` is a weird one, because it was never functional in pre-login mode, but was added to be ignored due to a bug in the Virgil client that sometimes sent it before logging in due to asynchronous code execution. It used to return a fake zone list with only `$Eden` before login, but now returns a `status: false` instead.

### 8.719.4 File

Defined in file `src/websockets.lisp`.

## 8.720 Tootsville::Infinity-Prompt-Reply

### 8.720.1 Function

Infinity-Prompt-Reply names a function, with lambda list (D USER RECIPIENT/S):

Accept a reply to a server-initiated prompt

Lisp PROMPT-REPLY = JSON promptReply

### 8.720.2 Usage

```
{ id: ID, reply: TOKEN }
```

### 8.720.3 Overview of Prompts

Server initiates prompt with:

```
{ "from" : "prompt",
  "id" : $ID,
  "label" : $LABEL,
  "label_en_US" : $LABEL,
  "title" : $TITLE,
  [ "attachUser" : $AVATAR_LABEL || "attachItem" : $ITEM_ID ] ,
  "msg" : $TEXT,
  "replies":
    { $TOKEN :
      { "label" : $BUTTON_LABEL,
        "label_en_US" : $BUTTON_LABEL,
        "type" : $BUTTON_TYPE },
      [ ... ]
    }
}
```

Where:

**\$ID** arbitrary string with no 0 (null byte) representing this question uniquely. This is not an user-visible string.

**\$LABEL** concatenated to the window title, but can be used to special-case / theme dialogs in future for certain purposes

**\$TITLE** dialog title

Only one of either “attachUser” or “attachItem” will be included. \$AVATAR\_LABEL is the full avatar label of the user/avatar to which the prompt should be attached — including “\$” and instance ID, if necessary — where \$ITEM\_ID is the room variable item ID for a placed item in the room.

\$TEXT = message text, may have n, will often need word-wrapping, and ideally might make use of scroll bars

The "replies" assoc-array is of arbitrary length, where the key to each item is a \$TOKEN, again an arbitrary string without 0 to represent this response uniquely. This is not an user-visible string.

`$BUTTON_LABEL` = the text to display. In future, the client may want to special-case specific text to use icons or something: e.g. "OK" will always be sent as precisely "OK" in English locale.

`$BUTTON_TYPE` = the type of the button for theming purposes only. This is from the enumerated set [ "aff" | "neg" | "neu" ];

`aff`        affirmative button, e.g. green button  
`neg`        negative button, e.g. red button  
`neu`        neutral button, e.g. purple button

To simplify future i18n/l10n efforts, the `$LABEL` and `$BUTTON_LABEL` will always be sent twice. The user's current language version will be in the "label" properties. The versions of those strings in the "en\_US" locale will always be in the "label\_en\_US" properties. For purposes of theming and such, the label\_en\_US properties should be considered; the "label" properties, however, should always be used in presentation to the end-user.

Example:

```
{ "from": "prompt",
  "status": "true",
  "id": "fountain/tootSquare/Ã¼³=?/x'deadbeef'",
  "label": "Fountain",
  "label_en_US": "Fountain",
  "title": "Make a Wish?",
  "msg": "Do you want to make a wish on the Toot Square fountain?",
  "replies":
  { "yes": { "label": "Make a Wish!",
            "label_en_US": "Make a Wish!",
            "type": "aff" },
    "no": { "label": "Not now",
           "label_en_US": "Not now",
           "type": "neg" }
  }
}
```

The client's response is a bit simpler:

```
{ "c": "promptReply", "d": { "id": $ID, "reply": $TOKEN } }
e.g.
{ "c": "promptReply",
  "d": { "id": "fountain/tootSquare/Ã¼³=?/x'deadbeef'",
        "reply": "yes" } }
```

As a special-case, for the reply only, the special `$TOKEN` of "close" should be sent if the user dismissed the dialog box with the close button.

I'd suggest that the GUI attach anonymous functions with the reply packets already constructed to the various dialog box controls at creation time, rather than trying to manage some queue of pending prompts.

To handle user expectations, it would be best to display the button in a "down" state until receiving the server's acknowledgement of the "promptReply" and disallow multiple-clicking in the window.

The server will respond with

```
{ "from": "promptReply", "status": "true", "id": $ID }
```

For debugging purposes, the server may reply with

```
{ from: "promptReply", "status": false, err: $ERR }
```

Where \$ERR will be a brief description of the problem.

`reply.notFound`

a reply button that was not a valid \$TOKEN from the "prompt" command nor the special case `close`.

`id.notFound`

a reply to a prompt that was not (recently) asked.

A prompt ID is not valid across sessions; pending prompts should be auto-closed on logout. Prompts can, however, remain active indefinitely, even across room joins.

#### 8.720.4 Canceling a prompt

Optional implementation: the server may cancel an outstanding prompt request by sending a packet with the following properties:

```
{ from: prompt
  status: true
  cancel: $ID }
```

Client applications may choose to dismiss the prompt automatically upon receiving such a packet. Failure to do so is not an error, however, later attempting to reply to a canceled prompt will return `status: false, err: id.notFound`. Clients must accept a cancelation packet silently if they do not process it.

#### 8.720.5 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.721 Tootsville::Infinity-Quiesce

### 8.721.1 Function

Infinity-Quiesce names a function, with lambda list (D TOOT R):

Quiesce Toot values to database for logout, or periodically as a backup.

Lisp QUIESCE = JSON quiesce

### 8.721.2 Usage

```
{ wt1: { course: { ... }, facing: RADIANS },
  d3: { ... },
  emotion: "EXPRESSION",
  world: "WORLD",
  latitude: LAT,
  longitude: LONG,
  altitude: ALT }
```

Facing may be provided as per Section 8.750 [TOOTSVILLE INTERPRET-FACING], page 1035.

A quiescent copy of the character information will be saved in a central database. Should the player lose connection and not successfully reconnect, eg. should the player quit by closing their browser altogether, or lose Internet connectivity, &c., the last quiesced form of their character will be restored when they reconnect.

Note that, as with the rest of the system, we are currently using `wt1` but are building up the infrastructure for `d3` walking in future. When both a `wt1` and a `d3` value exist, the `d3` value supersedes the `wt1`. `d` walking values must be encoded in `wt1` form, but `d` walking values are not expected in Tootsville V.

### 8.721.3 Status 200 OK

Upon success, the client whose status was saved is notified by a message of the form:

```
{ from: "quiesce",
  status: true }
```

### 8.721.4 Asynchronous periodic demands

From time to time, clients may be asked to update their quiescent state. When a client receives a message of the form:

```
{ from: "quiesce",
  status: false }
```

... they are expected to submit a quiesce message to the central servers.

### 8.721.5 File

Defined in file `src/infinity/new-commands-20.lisp`.



## 8.722 Tootsville::Infinity-Read-Map

### 8.722.1 Function

Infinity-Read-Map names a function, with lambda list (D U R):

Get the positions of badges and named locations on the map.

Lisp READ-MAP = JSON readMap

### 8.722.2 Usage

This command requires no parameters.

### 8.722.3 Status 200 OK

This returns two lists, a list of named places, and a list of badge places. These make up the map of Tootanga.

```
{ from: "readMap",
  status: true,
  spots: [ TootSquare: [ 0, 0, 0, "CHOR", "Toot Square" ], ... ],
  badges: [ ... ] }
```

The lists `spots` and `badges` each consist of a set of monikers or labels as keys.

The values of each key on `spots` are the latitude, longitude, altitude, world, and UI label for that spot.

The values of each key on `badges` are only a latitude, longitude, altitude, and world. The moniker itself represents the badge.

An SVG graphic for each badge should be found on Jumbo in the form <https://jumbo.tootsville.org/Assets/Badges/5/BADGE-NAME.svg>.

### 8.722.4 Overview of Spots and Badges

A Spot, or a Named Place, associates a moniker with a text label and a position on the map. These can be used by certain operator commands, or by users, and basically represent what used to be rooms in earlier versions of Tootsville. We are trying to enforce the use of the name “spot” to mean a named place to limit the relative confusion with Places found within the game world, such as cobblestone paths.

A Badge represents a graphic that appears on the Map App on the Tootnix Mobile phone. These are usually magnets indicating a special event, or a visit by a special character, at that location. Little Reminders mice will be used to draw the players’ attention to these badges in order to garner attention to special events.

### 8.722.5 File

Defined in file `src/infinity/new-commands-20.lisp`.

## 8.723 Tootsville::Infinity-Remove-From-List

### 8.723.1 Function

Infinity-Remove-From-List names a function, with lambda list (D USER RECIPIENT/S):

Remove someone from a buddy list or ignore list.

Lisp REMOVE-FROM-LIST = JSON removeFromList

### 8.723.2 Usage

To drop a buddy from the buddy list:

```
{ buddy: "user-name" }
```

To attend to someone who had previously been ignored:

```
{ ignore: "name" }
```

### 8.723.3 Status 200 OK

The user was removed from the buddy list or ignore list.

```
{ from: "removeFromList",
  status: true,
  buddy: "buddy-name" }
```

```
{ from: "removeFromList",
  status: true,
  ignore: "ignored-name" }
```

### 8.723.4 Status 404 Not Found

The user name given was not found.

```
{ from: "removeFromList",
  status: false,
  err: "login.notFound" }
```

### 8.723.5 Status 412 Precondition Failed

An attempt was made to remove someone from the buddy or ignore list who was not on that list.

```
{ from: "removeFromList",
  status: false,
  err: "notOnList" }
```

### 8.723.6 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.724 Tootsville::Infinity-Report-Bug

### 8.724.1 Function

Infinity-Report-Bug names a function, with lambda list (D USER RECIPIENT/S):

This method allows the client to “phone home” to report a bug.

Lisp REPORT-BUG = JSON reportBug

The bug report itself is just a giant string embedded in the “bug” element, but a “cause” element will be treated as the subject. Note that the bug report — like all JSON input — will be cut off at a certain limit (typically 4KiB), so it’s most helpful to keep it short & sweet: Typically, this should be something like a single stack backtrace (with as much detail as possible), rather than a complete log trace or something.

The suggested usage is to include the exception itself as “cause,” the backtrace up to a maximum of 1KiB, a log backtrace up to its last 1KiB as “bug,” and as much machine-formatted system information as possible in the “info” object. Fields of “info”

As many fields as possible, limit the contents to a reasonable length though. . .

Note that the keys listed are strings, so e.g.:

```
info ["navigator.language"] = navigator.language;
info ["navigator.product"] = navigator.product;
```

ActionScript example:

```
var info:Object = {
    "flash.sys.ime": flash.system.System.ime,
    "flash.sys.totalMemory": flash.system.System.totalMemory,
    "flash.sys.useCodePage": flash.system.System.useCodePage
};
// imperfect but close
for ( var key in flash.system.Capabilities ) {
    info["flash.sysCap." + key] = flash.system.Capabilities[key];
}
```

```
‘navigator.language’
    JavaScript: navigator.language
```

```
‘navigator.product’
    JavaScript: navigator.product
```

```
‘navigator.appVersion’
    JavaScript: navigator.appVersion
```

```
‘navigator.platform’
    JavaScript: navigator.platform
```

```
‘navigator.vendor’
    JavaScript: navigator.vendor
```

```
‘navigator.appCodeName’
    JavaScript: navigator.appCodeName
```

```
‘navigator.cookieEnabled’
    JavaScript: navigator.cookieEnabled
```

```
'navigator.appName'  
    JavaScript: navigator.appName  
  
'navigator.productSub'  
    JavaScript: navigator.productSub  
  
'navigator.userAgent'  
    JavaScript: navigator.userAgent  
  
'navigator.vendorSub'  
    JavaScript: navigator.vendorSub  
  
'screen.height'  
    JavaScript: screen.height;  
    ActionScript: flash.system.Capabilities.screenResolutionX  
  
'screen.width'  
    JavaScript: screen.width;  
    ActionScript: flash.system.Capabilities.screenResolutionY  
  
'screen.availHeight'  
    JavaScript: screen.availHeight;  
    ActionScript: flash.display.Stage.fullScreenHeight  
  
'screen.availWidth'  
    JavaScript: screen.availWidth;  
    ActionScript: flash.display.Stage.fullScreenWidth  
  
'window.outerHeight'  
    JavaScript: window.outerheight note case  
  
'window.outerWidth'  
    JavaScript: window.outerwidth note case  
  
'window.innerHeight'  
    JavaScript: window.innerheight note case  
  
'window.innerWidth'  
    JavaScript: window.innerWidth note case  
  
'window.windowName'  
    JavaScript: the window.name property of the highest parent of this window  
    (frame); e.g.  
        var topWindow = window.parent;  
        for (; topWindow.parent != topWindow;  
            topWindow = topWindow.parent)  
        ;  
        info ["window.windowName"] = topWindow.name;  
  
'flash.sys.totalMemory'  
    ActionScript: flash.system.System.totalMemory  
  
'flash.sys.ime'  
    ActionScript: flash.system.System.ime
```

```
'flash.sys.useCodePage'  
    ActionScript: flash.system.System.useCodePage  
  
'flash.sysCap.avHardwareDisable'  
    ActionScript: flash.system.Capabilities.avHardwareDisable  
  
'flash.sysCap.hasAccessibility'  
    ActionScript: flash.system.Capabilities.hasAccessibility  
  
'flash.sysCap.hasAudio'  
    ActionScript: flash.system.Capabilities.hasAudio  
  
'flash.sysCap.hasAudioEncoder'  
    ActionScript: flash.system.Capabilities.hasAudioEncoder  
  
'flash.sysCap.hasEmbeddedVideo'  
    ActionScript: flash.system.Capabilities.hasEmbeddedVideo  
  
'flash.sysCap.hasIME'  
    ActionScript: flash.system.Capabilities.hasIME  
  
'flash.sysCap.hasMP3'  
    ActionScript: flash.system.Capabilities.hasMP3  
  
'flash.sysCap.hasPrinting'  
    ActionScript: flash.system.Capabilities.hasPrinting  
  
'flash.sysCap.hasScreenBroadcast'  
    ActionScript: flash.system.Capabilities.hasScreenBroadcast  
  
'flash.sysCap.hasScreenPlayback'  
    ActionScript: flash.system.Capabilities.hasScreenPlayback  
  
'flash.sysCap.hasStreamingAudio'  
    ActionScript: flash.system.Capabilities.hasStreamingAudio  
  
'flash.sysCap.hasStreamingVideo'  
    ActionScript: flash.system.Capabilities.hasStreamingVideo  
  
'flash.sysCap.hasTLS'  
    ActionScript: flash.system.Capabilities.hasTLS  
  
'flash.sysCap.hasVideoEncoder'  
    ActionScript: flash.system.Capabilities.hasVideoEncoder  
  
'flash.sysCap.isDebugger'  
    ActionScript: flash.system.Capabilities.isDebugger  
  
'flash.sysCap.isEmbeddedInAcrobat'  
    ActionScript: flash.system.Capabilities.isEmbeddedInAcrobat  
  
'flash.sysCap.language'  
    ActionScript: flash.system.Capabilities.language  
  
'flash.sysCap.localFileReadDisable'  
    ActionScript: flash.system.Capabilities.localFileReadDisable
```

```
'flash.sysCap.manufacturer'  
    ActionScript: flash.system.Capabilities.manufacturer  
  
'flash.sysCap.os'  
    ActionScript: flash.system.Capabilities.os  
  
'flash.sysCap.pixelAspectRatio'  
    ActionScript: flash.system.Capabilities.pixelAspectRatio  
  
'flash.sysCap.playerType'  
    ActionScript: flash.system.Capabilities.playerType  
  
'flash.sysCap.screenColor'  
    ActionScript: flash.system.Capabilities.screenColor  
  
'flash.sysCap.screenDPI'  
    ActionScript: flash.system.Capabilities.screenDPI  
  
'flash.sysCap.version'  
    ActionScript: flash.system.Capabilities.version  
  
'flash.displayState'  
    ActionScript: if flash.display.Stage.displayState == FULL_SCREEN_INTERACTIVE,  
    then "fullScreen"; for NORMAL, return "window".  
  
'flash.frameRate'  
    ActionScript: flash.display.Stage.frameRate  
  
'flash.quality'  
    ActionScript: flash.display.Stage.quality  
  
'flash.scaleMode'  
    ActionScript: flash.display.Stage.scaleMode  
  
  
// ActionScript example  
function systemReport:Object () {  
    return {  
        "screen": {  
            "height": flash.system.Capabilities.screenResolutionX,  
            "width": flash.system.Capabilities.screenResolutionY,  
            "availHeight": flash.display.Stage.fullScreenHeight,  
            "availWidth": flash.display.Stage.fullScreenWidth,  
        },  
        "flash": {  
            "sys": {  
                "totalMemory": flash.system.System.totalMemory,  
                "ime": flash.system.System.ime,  
                "useCodePage": flash.system.System.useCodePage,  
            },  
            "sysCap": {  
                "avHardwareDisable": flash.system.Capabilities.avHardwareDisable,  
                "hasAccessibility": flash.system.Capabilities.hasAccessibility,  
            }  
        }  
    }  
}
```

```

"hasAudio": flash.system.Capabilities.hasAudio,
"hasAudioEncoder": flash.system.Capabilities.hasAudioEncoder,
"hasEmbeddedVideo": flash.system.Capabilities.hasEmbeddedVideo,
"hasIME": flash.system.Capabilities.hasIME,
"hasMP3": flash.system.Capabilities.hasMP3,
"hasPrinting": flash.system.Capabilities.hasPrinting,
"hasScreenBroadcast": flash.system.Capabilities.hasScreenBroadcast,
"hasScreenPlayback": flash.system.Capabilities.hasScreenPlayback,
"hasStreamingAudio": flash.system.Capabilities.hasStreamingAudio,
"hasStreamingVideo": flash.system.Capabilities.hasStreamingVideo,
"hasTLS": flash.system.Capabilities.hasTLS,
"hasVideoEncoder": flash.system.Capabilities.hasVideoEncoder,
"isDebugger": flash.system.Capabilities.isDebugger,
"isEmbeddedInAcrobat": flash.system.Capabilities.isEmbeddedInAcrobat,
"language": flash.system.Capabilities.language,
"localFileReadDisable": flash.system.Capabilities.localFileReadDisable,
"manufacturer": flash.system.Capabilities.manufacturer,
"os": flash.system.Capabilities.os,
"pixelAspectRatio": flash.system.Capabilities.pixelAspectRatio,
"playerType": flash.system.Capabilities.playerType,
"screenColor": flash.system.Capabilities.screenColor,
"screenDPI": flash.system.Capabilities.screenDPI,
"version": flash.system.Capabilities.version
},
"displayState": ( flash.display.Stage.displayState ==
                    FULL_SCREEN_INTERACTIVE ?
                    "fullScreen" : "window" ),
"frameRate": flash.display.Stage.frameRate,
"quality": flash.display.Stage.quality,
"scaleMode": flash.display.Stage.scaleMode
}
};
}

```

jso - Must contain a single string attribute named “bug.” Should contain an attribute named “info” with system information key-value pairs (see above). May also have a subject of “cause” as a string.

u - The user reporting the bug.

## 8.724.2 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.725 Tootsville::Infinity-Report-User

### 8.725.1 Function

Infinity-Report-User names a function, with lambda list (D USER RECIPIENT/S):

Report an user to the moderator(s) on duty for breaking a rule

Lisp REPORT-USER = JSON reportUser

{ userName = user to be reported }

### 8.725.2 File

Defined in file src/infinity/legacy-commands.lisp.



## 8.726 Tootsville::Infinity-Request-Buddy

### 8.726.1 Function

Infinity-Request-Buddy names a function, with lambda list (D USER RECIPIENT/S):

Request adding a user to your buddy list (mutual-add) using the notification-based system

```
Lisp REQUEST-BUDDY = JSON requestBuddy  
(Added in 1.1)
```

### 8.726.2 Usage

```
{ buddy: LOGIN }
```

### 8.726.3 Example

```
{ buddy: "catville" }
```

### 8.726.4 Changes from 1.0 to 1.1

The old system allowed users to simply add anyone to their buddy list; cv. Section 8.680 [TOOTSVILLE INFINITY-ADD-TO-LIST], page 938. The new system requires mutually confirmed adding. AKA the Twitter vs. Facebook mechanisms.

### 8.726.5 New in 1.1

This was new in Romance 1.1

### 8.726.6 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.727 Tootsville::Infinity-Send-Mail-Message

### 8.727.1 Function

Infinity-Send-Mail-Message names a function, with lambda list (D U R):

Send an in-game SMS message.

Lisp SEND-MAIL-MESSAGE = JSON sendMailMessage

### 8.727.2 Usage

```
{ ( to: "RECIPIENT" | toList: [ "RECIPIENT", ... ] ),
  [ subject: "" ],
  body: "BODY",
  [ uuid: UUID-STRING ] )
```

`subject` is optional, and should be omitted in 2.0. Non-empty subjects will return an error.

`uuid` is optional but recommended. It allows the client to track when a message has been sent.

See Section 8.1119 [TOOTSVILLE SEND-SMS-MESSAGE], page 1404, for the underlying implementation.

### 8.727.3 Examples

```
{ to: "shader", subject: "", body: "Hello there!" }
```

```
{ toList: [ "catville", "pil" ], body: "Howdy" }
```

Input: `subject` (must be blank); `to`, the Toot name to whom to send the text; and `body` of the message.

Rather than `to`, the user can send `toList` with an object, the keys of which are ignored, the values of which are Toot names.

### 8.727.4 Changes from 1.2 to 2.0

Subjects are no longer supported. `subject` must be absent, null, or "".

`uuid` is a new option.

Message length is now measured in Unicode characters, not bytes.

### 8.727.5 Formerly Proprietary Extension

This command was formerly a proprietary extension for Tootsville.com and has now been re-created for the AGPL version of Romance.

### 8.727.6 200 OK

The “SMS” message was sent.

```
{ from: "sendMailMessage", status: true }
{ from: "sendMailMessage", status: true,
  uuid: "5047F44E-8B1D-4B8A-9EC6-4E1D6E1653AD" }
```

If the client supplied an UUID, it will be returned, allowing the client to identify which of potentially many SMS messages was sent.

Sending does not imply that the message was received or read by the destination user.

### 8.727.7 400 Bad Request

If an UUID was supplied with the request, the response will echo it.

subject must be absent, null, or ""

```
{ from: "sendMailMessage", status: false,  
  error: "Subject is not allowed. Please leave subject blank." }
```

Exactly one of to or toList must be specified

```
{ from: "sendMailMessage", status: false,  
  error: "Message has no destination.",  
  uuid: "E6726651-703D-41FC-8484-E59EADDEE7EA0" }
```

body may not be empty

### 8.727.8 413 Payload Too Large

body can be at most 1,024 (Unicode) characters (not bytes).

```
{ from: "sendMailMessage", status: false,  
  error: "Message too long. Try a message with less than 1,000 characters." }
```

### 8.727.9 Formerly Proprietary Extension

This command was formerly a proprietary extension for Tootsville.com and has now been re-created for the AGPL version of Romance.

### 8.727.10 File

Defined in file src/infinity/tootsville-commands.lisp.

## 8.728 Tootsville::Infinity-Send-Out-Of-Band-Message

### 8.728.1 Function

Infinity-Send-Out-Of-Band-Message names a function, with lambda list (D USER RECIPIENT/S):

Send an arbitrary JSON packet to another user, or all of the users

Lisp SEND-OUT-OF-BAND-MESSAGE = JSON sendOutOfBandMessage

Out of the band of communications.

This is neither a public nor a private message in the chat context: just some additional data that is being provided.

```
{ sender: sender, from: outOfBand, status: true, body: {JSON} }
```

Adds "roomTitle" to body if body contains "room" and title can be determined

Add "sendRoomList": "true" to give the user an updated room list as well. (Necessary for invitations to new rooms.) Inviting to houses ...

```
initUserRoom { room: 0, autoJoin: false }
```

```
{ from: initUserRoom, status: true, moniker: ROOM-MONIKER } ** OK
```

```
=> { from: initUserRoom, status: false, err: exists, moniker: ROOM-MONIKER } ** OK
```

```
=> { from: initUserRoom, status: false, err: showFirstRun } ** ERR (player does not have that room)
```

```
sendOutOfBandMessage { to: USER-LOGIN, body: { locType: "house", type: "invite", room: MONIKER } }
```

```
{ from: outOfBand, sender: YOUR-LOGIN, status: true, body: { locType: "house", type: "invite", room: MONIKER, roomTitle: USER-VISIBLE-NAME } }
```

for user houses, roomTitle will be like "BlackDaddyNerd's House"

Parameters:

jso - To send to one user: { to: userName, body: {JSON} }, or to broadcast to the entire room: { toRoom: true, body: {JSON} }

u - The sender of the out-of-band-message

room - The room in which the sender is standing. Necessary for the toRoom version of this method.

Throws:

org.json.JSONException - Thrown if the data cannot be interpreted from the JSON objects passed in, or conversely, if we can't encode a response into a JSON form

### 8.728.2 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.729 Tootsville::Infinity-Server-Time

### 8.729.1 Function

Infinity-Server-Time names a function, with lambda list (D U R):

Accept the client's notification of a server-time adjustment.

Lisp SERVER-TIME = JSON serverTime

This is used to compute the client's round-trip lag time.

### 8.729.2 Usage

```
{ serverTime: LONG milliseconds since Unix epoch }
```

### 8.729.3 Example

```
{ serverTime: 1589850683000 }
```

### 8.729.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.730 Tootsville::Infinity-Set-Avatar-Color

### 8.730.1 Function

Infinity-Set-Avatar-Color names a function, with lambda list (D USER RECIPIENT/S):

Set the avatar base and extra (pad) colours for the given user.

Lisp SET-AVATAR-COLOR = JSON setAvatarColor

This function is no longer available. Doodle must change the avatar's color now.

### 8.730.2 Romance 1.1 instructions

Colour numbers are given in X'RRGGBB' form as an integer — to compute one from byte (0..255) RGB values, do ( red << 16 & green << 8 & blue )

Parameters: jso - { "base": (colour number), "extra": (colour number) } u - The user whose avatar colours are being set room - The room in which the user is standing Throws:

org.json.JSONException - Thrown if the data cannot be interpreted from the JSON objects passed in, or conversely, if we can't encode a response into a JSON form

SQLException - if the palettes can't be loaded

### 8.730.3 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.731 Tootsville::Infinity-Set-Furniture

### 8.731.1 Function

Infinity-Set-Furniture names a function, with lambda list (D USER RECIPIENT/S):

Set or change a furniture item.

Lisp SET-FURNITURE = JSON setFurniture

There are 3 distinct forms in which this command can be used.

- item** To add a piece of furniture to the area, send a packet with the following data:
- item** The item's class number — WRITEME . . . You must have an item of this type in your inventory.
  - x, y, z** The position at which to place the item
  - facing** The angle (in radians) of the facing of this item. Alternatively, for backward compatibility with 1.2, the facing direction can be specified as N, NE, E, SE, S, SW, W, or NW (case-insensitive). See Section 8.750 [TOOTSVILLE INTERPRET-FACING], page 1035.
- slot** To position or reposition a particular item by its UUID, send a packet like:
- slot** The item's UUID. This item must be in your inventory.
  - x, y, z** The position at which to (re)position the item
  - facing** The angle (in radians) of the facing of this item. Alternatively, for backward compatibility with 1.2, the facing direction can be specified as N, NE, E, SE, S, SW, W, or NW (case-insensitive). See Section 8.750 [TOOTSVILLE INTERPRET-FACING], page 1035,
- remove** Remove a furniture item from the scene, putting it back into your inventory. The packet keys will look like:
- remove** This value must be **true**.
  - slot** The UUID of the item to remove from the scene.

### 8.731.2 Romance 1.2 instructions

To add a structural item to the room, put `item: 123` without anything else. To place furniture on the floor, also add attributes `x`, `y`, and `facing`.

To change furniture, replace `item:` with `slot:` (to avoid ambiguities about “which chair”)

To remove an item from the room, send `{ slot: 123, remove: true }`

Parameters:

`jso - { slot: #, x: #, y: #, facing: $ } or { item: #, x: #, y: #, facing: $ } or { slot: #, remove: true }`

### 8.731.3 Changes from 1.2 to 2.0

- z position is required
- structural items no longer exist
- slot is the item's UUID
- facing can be specified as an arbitrary angle, rather than just the eight cardinal directions.

### 8.731.4 200 OK

WRITEME

### 8.731.5 400 Error in parameters

This error is thrown if the parameters are not in one of ; the accepted formats

WRITEME

### 8.731.6 File

Defined in file src/infinity/legacy-commands.lisp.



## 8.732 Tootsville::Infinity-Set-Room-Var

### 8.732.1 Function

Infinity-Set-Room-Var names a function, with lambda list (D USER RECIPIENT/S):

Set a room variable or set of room variables.

Lisp SET-ROOM-VAR = JSON setRoomVar

There are no longer room variables (as such) in Romance 2.0. However, some of them can be fake-set to actually alter some underlying facts of the system, by a Builder Toot. ;

UNIMPLEMENTED

WRITEME

### 8.732.2 Usage

jso - key-value pair(s) for room variable(s) to be set

WRITEME

### 8.732.3 Example

WRITEME

### 8.732.4 Changes from 1.2 to 2.0

WRITEME

### 8.732.5 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.733 Tootsville::Infinity-Set-User-Var

### 8.733.1 Function

Infinity-Set-User-Var names a function, with lambda list (D USER RECIPIENT/S):

Set “User Variables”

Lisp SET-USER-VAR = JSON setUserVar

### 8.733.2 Usage

```
{ "KEY": "VALUE" [ ... ] }
```

```
{ d: "D-String" }
```

```
{ wtl: course: { COURSE }, facing: FACING }
```

```
{ d3: course: { COURSE } }
```

```
{ xpr: "expression" }
```

```
{ sN: "D-String" }
```

```
{ shotN: course: { COURSE }, facing: FACING }
```

This is a legacy-type method, which is provided for the convenience of client implementors.

### 8.733.3 Example

```
{ c: "setUserVar",
  { d: "100~100~200~200~NE~6029604401000",
    s0: "100~100~300~300~NE~6029604401000",
    xpr: "smile" } }
```

### 8.733.4 Changes from 1.2 to 2.0

Historically, arbitrary attributes could be attached to a user in the game, as transient values that remained as long as that user was connected. Thus, any key:value pair could be “advertised” by a user by posting them to this method.

### 8.733.5 Available Attributes (2.0)

In Romance II, only the following “user variable” key names are actually supported:

- d This is the legacy “d” string for purposes of positioning and guiding a character. Since it was designed for a 2-dimensional space, the coordinate space is treated as (x,z) rather than (x,y) if there is no “z” coordinate given. Since Romance II clients are expected to use wtl or d3 packets only, this will be translated into a wtl course and then transmitted.  
See Section 8.744 [TOOTSVILLE INFINITY-WTL], page 1028, for a discussion of its structure.
- wtl See Section 8.744 [TOOTSVILLE INFINITY-WTL], page 1028, for the structure of this linear course.

**d3** This is an experimental format not yet used in Romance 2.0. It will support more complex path descriptions.

**xpr** This sets the player's expression (on their face); not yet supported in Tootsville V.

**sN**  
This is a shot position in **d** form, where *N* is an arbitrary unique identifier chosen by the client. See Section 8.734 [TOOTSVILLE INFINITY-SHOOT], page 1014, for another way to provide this data.

**shotN**  
This is a shot position in **wt1** form, where *N* is an arbitrary unique identifier chosen by the client.

Any other KEY value will result in an error.

### 8.733.6 200 OK

When all keys are set successfully, this will return with a packet like

```
{ from: "setUserVar",
  status: true,
  set: [ "key", ... ] }
```

When some keys could not be set, they will be listed separately

```
{ from: "setUserVar",
  status: true,
  set: [ "key", ... ],
  unset: [ "key", ... ] }
```

### 8.733.7 400 Illegal

When no key is from the set of supported keys, an error is returned:

```
{ from: "setUserVar",
  status: false,
  unset: [ "key", ... ] }
```

### 8.733.8 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.734 Tootsville::Infinity-Shoot

### 8.734.1 Function

Infinity-Shoot names a function, with lambda list (D U R):

Fire a shot from a projectile device.

Lisp SHOOT = JSON shoot

UNIMPLEMENTED

The projectile device ITEM must be capable of firing a projectile; this includes having sufficient energy (ammunition) to do so.

Projectiles are currently UNIMPLEMENTED.

### 8.734.2 Usage

```
{ c: "shoot",
  d: { i: ITEM,
       course: COURSE,
       facing: FACING } }
```

Facing is interpreted by Section 8.750 [TOOTSVILLE INTERPRET-FACING], page 1035.

### 8.734.3 Example

WRITEME

### 8.734.4 See also

See also the Section 8.733 [TOOTSVILLE INFINITY-SET-USER-VAR], page 1012, command for an alternative way to promulgate shots.

### 8.734.5 File

Defined in file src/infinity/new-commands-20.lisp.

## 8.735 Tootsville::Infinity-Spawn-Zone

### 8.735.1 Function

Infinity-Spawn-Zone names a function, with lambda list (D USER RECIPIENT/S):

Spawn an additional server peer pairing.

Lisp SPAWN-ZONE = JSON spawnZone

UNIMPLEMENTED

### 8.735.2 Implementation in 2.0

We no longer have zones, but we can have server pairings.

This is used to establish a new server pairing ...

WRITEME

### 8.735.3 Changes from 1.2 to 2.0

WRITEME

### 8.735.4 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.736 Tootsville::Infinity-Speak

### 8.736.1 Function

Infinity-Speak names a function, with lambda list (D USER RECIPIENT/S):

The user speaks SPEECH at volume VOL in public.

Lisp SPEAK = JSON speak

Handle speech by the user.

Speech is public to all users in an area.

### 8.736.2 Usage

```
{ c: "speech",
  d: { speech: "text to be spoken",
       vol: ( "shout" | "talk" | "whisper" ) } }
```

key — WRITEME — optional — currently ignored

vol — Volume is one of `talk`, `shout`, or `whisper`. The default is always `talk`. `vol` is optional.

### 8.736.3 Speech filtering

There are two kinds of filtering on text: foul language, and obnoxious typing.

Foul language filtering occurs when there are children or sensitive players nearby (blacklist), and in all cases for certain stopwords (redlist).

See Section 8.178 [TOOTSVILLE CASSANDRA-FILTER], page 432.

Obnoxious typing filtering occurs all the time, and undoes a couple of things that are — well, just plain obnoxious.

- SPEECH IN ALL CAPS is converted into lower-case; if it was meant to be whispered, it will instead be spoken (`talk`); if it was meant to be spoken, it will instead be shouted. Shouted text in all caps remains shouted (but is still in lower-case).
- Sentences with lots of punctuation!! are fixed; aside from ellipses, no repeated punctuation is preserved.

See Section 8.179 [TOOTSVILLE CASSANDRA-OBNOXIOUS-FILTER], page 433.

### 8.736.4 Special character handling

The first character of the speech can turn it into a special command of some kind.

~

Commands beginning with ~ should be handled by the client. A conforming client should never forward any command beginning with ~ to the server.

# Server commands begin with # (sharp sign / octothorpe / hash sign). A server command is any unary function in the **Tootsville-User** package. See Section 8.946 [TOOTSVILLE PARSE-OPERATOR-COMMAND], page 1231, for details.

@ @-messages are whispered directly to the named character, if they are located somewhere in the nearby area. For example, @Catville Hello! will whisper the phrase Hello! to only the player Catville.

/ Emotes begin with /. Emotes set the expression of the character to one of a predefined list of expressions, or display an emoji speech balloon.

Why these specific emojis? Backward compatibility. This list of emotes was inherited from Tootsville IV.

The emoji items might be replaced with more detailed animations in the future. A few of the emotes actually have even more complex behavior, as noted in the index below.

smile	The expression on the character's face should change to a smile.
frown	The expression on the character's face should change to a frown.
wink	The character should make an exaggerated wink.
sick	The expression on the character's face should change to disgust.
whoa	The expression on the character's face should change to surprise.
cool	An emoji of a smiling face wearing sunglasses.
cheese	The expression on the character's face should change to a smile with tongue stuck out.
angry	The expression on the character's face should change to anger.
silly	The expression on the character's face should change to a silly face.
sleep	The character's face should look as though they are sleeping. Also, a special "Zzz" graphic should appear over their head.
meh	The expression on the character's face should change to disinterest.
cry	The character should begin to cry.
pizza	An emoji of a pizza.
burger	An emoji of a burger.
hotdog	An emoji of an hot dog.
fries	An emoji of a pack of French fries.
drink	An emoji of a glass of an unidentified beverage.
icecream	An emoji of ice cream.
cake	An emoji of a slice of cake.
game	WRITEME
dice	An emoji will be spoken showing a single 6-sided die; however, the number of pips shown (1-6) will be random.
coin	An emoji will be spoken showing a coin; however, whether that coin shows as head or tails will be random.
heart	A heart emoji
broken	A broken heart emoji

<code>rps</code>	An emoji will appear with a random selection from: rock, paper, scissors.
<code>music</code>	A musical note emoji.
<code>rainbow</code>	A rainbow emoji.
<code>lol</code>	The character's expression should change to laughter.
<code>rain</code>	A raincloud emoji.
<code>huh</code>	The character's expression should change to confusion.

`% _ ^ |`

These characters are reserved for future use. You cannot speak a line beginning with them.

`? !`

For convenience of Spanish speakers, sentences beginning with `?` or `!` are converted into `¿` and `¡`.

UNIMPLEMENTED

### 8.736.5 Special commands

`,credits` Speaking `,credits` will send the server's credits as an admin message.

`,disconnect`

Speaking `,disconnect` will immediately drop the client's connection without ceremony; it's used for testing the auto-reconnection code.

`,dumpthreads`

This will log all active threads to the server log. Note that it is not an operator command in this context, but it is identical to the operator command `#dumpthreads`.

### 8.736.6 Changes from 1.2 to 2.0

WRITEME

### 8.736.7 File

Defined in file `src/infinity/legacy-commands.lisp`.



## 8.737 Tootsville::Infinity-Stamp-Passport

### 8.737.1 Function

Infinity-Stamp-Passport names a function, with lambda list (D U R):

Stamp the Toot's passport

Lisp STAMP-PASSPORT = JSON stampPassport

Passports are not currently implemented in Tootsville V, but will be returning.

See Section 8.1158 [TOOTSVILLE STAMP-TOOT-PASSPORT], page 1443.

### 8.737.2 Usage

```
{ room: "SPOT-MONIKER" }
```

### 8.737.3 Example

```
{ room: "tootSquare" }
```

### 8.737.4 Changes from 1.2 to 2.0

Passports stamps are temporarily unavailable.

`room` was previously a room moniker, but will now be a “spot” moniker of a Spot in the game world. Despite the change, the key name remains `room`.

### 8.737.5 Formerly Proprietary Extension

This command was formerly a proprietary extension for Tootsville.com and has now been re-created for the AGPL version of Romance.

### 8.737.6 File

Defined in file `src/infinity/tootsville-commands.lisp`.

## 8.738 Tootsville::Infinity-Start-Event

### 8.738.1 Function

Infinity-Start-Event names a function, with lambda list (D USER RECIPIENT/S):

Attempt to begin a Quaestor Event. Might return an error.

Lisp START-EVENT = JSON startEvent

The bulk of the actual work is in Section 8.1038 [TOOTSVILLE QUAESTOR-START-EVENT], page 1323, q.v.

### 8.738.2 What is a “Quaestor Event”?

Events, in the context of this function, are transactions between a player and the world. These transactions might yield items or currency (peanuts, or fairy dust), so they have to be proxied through the central servers, because we can't ultimately trust the users not to just tap Control+Shift+K and try something like `Tootsville.Game.addPeanuts(1000000)`. (Note, that will — obviously — not work, because this function exists.)

So, there are a few basic types of events, in general:

- Magic fountains
- Shops
- Secrets and treasures
- Minigames

Each of these works a little differently.

### 8.738.3 Usage

The basic data element is a `moniker`, which is typically the representation of a particular item in the world which is the focus of the event. In the current usage (Tootsville V/Romance 2), this will be an UUID.

```
{ c: "startEvent", d: { moniker: "moniker" } }
```

### 8.738.4 Responses

There are several possible responses.

#### 8.738.4.1 Event already completed

```
{ from: "startEvent",
  status: false,
  alreadyDone: true,
  err: "event.alreadyDone",
  error: "User-visible error message"
  moniker: "moniker" }
```

Some events cannot be started more than once by the same character, or more than once within a certain period of time, or more than once by the same character within a certain period of time. This is a simple rejection; there is not inherently any explanation to the client of the circumstances — in particular, the client is not informed when (or by whom) the event can be fired again.

### 8.738.4.2 Event started successfully

```
{ from: "startEvent",
  status: true,
  eventID: "ID" }
```

This is the short form. It means that the event can be started, and the caller had better know what to do about it; typically, that will only be to turn around and immediately call Section 8.689 [TOOTSVILLE INFINITY-END-EVENT], page 949, with the provided event ID.

### 8.738.4.3 Event requires a download to begin

```
{ from: "startEvent",
  status: true,
  eventID: "ID",
  filename: "blah.swf",
  asVersion: ( 2 | 3 ) }
```

This form is archaic and won't be returned right now, but is included for comparison — and to make the modern long form make sense by comparison.

```
{ from: "startEvent",
  status: true,
  eventID: "ID",
  filename: "blah.js",
  function: "foo",
  asVersion: "html5" }
```

This is the modern long form. The client is expected to:

- Download `blah.js`
- Call the function `Tootsville.Event["foo"](eventID)` — that is, literally, look in the global object `Tootsville.Event` for a function named `foo` — with the event ID as its parameter.

In other words,

```
Tootsville.Event [ datagram.function ] ( datagram.eventID );
```

The code in `blah.js` is required to use the opportunistic object-as-namespace initialization of the form:

```
if (!('Tootsville' in window))
  { Tootsville = { Event: { Foo: {} } } }; }
```

```
if (!('Event' in Tootsville))
  { Tootsville.Event = { Foo: {} } };
```

```
if (!('Foo' in Tootsville.Event))
  { Tootsville.Event.Foo = {} }; }
```

```
Tootsville.Event.foo = function (eventID) { ... };
```

```
Tootsville.Event.Foo.otherMethod = function ( ... ) { ... };
```

See the front-end documentation for more details on the coding style used.

### 8.738.5 Error response

```
{ from: "startEvent",  
  status: false,  
  err: "error code",  
  error: "User-visible error message" }
```

The error code can be one of:

```
eventType.notFound  
    The moniker passed was invalid.
```

### 8.738.6 Ending an event

This event is now open, and will remain open until it has been completed or canceled using Section 8.689 [TOOTSVILLE INFINITY-END-EVENT], page 949, q.v.

## 8.738.7 Quaestor Events in Detail

### 8.738.7.1 Magic Fountains

WRITEME

### 8.738.7.2 Shops

WRITEME

### 8.738.7.3 Secrets and Treasures

WRITEME

### 8.738.7.4 Minigames

WRITEME

## 8.738.8 Changes from 1.2 to 2.0

WRITEME

### 8.738.9 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.739 Tootsville::Infinity-Stats**

### **8.739.1 Function**

Infinity-Stats names an undocumented function, with lambda list NIL.

### **8.739.2 File**

Defined in file `src/infinity/infinity.lisp`.

## 8.740 Tootsville::Infinity-Toot-List

### 8.740.1 Function

Infinity-Toot-List names a function, with lambda list (D U RECIPIENT/S):

Enumerates all Toots owned by the user.

Lisp TOOT-LIST = JSON tootList

### 8.740.2 Usage

This command requires no parameters.

### 8.740.3 200 OK

Returns an object with `status: true`, `from: "tootList"`, and a key `toots` under which is the list of Toots owned by the user. Each Toot object is as per Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534, q.v.

```
{ from: "tootList",  
  status: true,  
  toots: [ { INFO }, ... ] }
```

See Section 8.1253 [TOOTSVILLE TOOT-LIST-MESSAGE], page 1540.

### 8.740.4 File

Defined in file `src/infinity/new-commands-20.lisp`.

## 8.741 Tootsville::Infinity-Use-Equipment

### 8.741.1 Function

Infinity-Use-Equipment names a function, with lambda list (D USER RECIPIENT/S):

The player wishes to use a piece of equipment on a particular item or place.

Lisp USE-EQUIPMENT = JSON useEquipment

### 8.741.2 Usage

```
{ t: ( 1 | 2 ),
  x: X, y: Y, z: Z }
```

or

```
{ t: ( 1 | 2 ),
  on: "ITEM-OR-CHARACTER-UUID",
  [ of: ( "item" | "char" ) ] }
```

The `t` number indicates whether the user's currently-selected primary item (i.e. the item equipped in their trunk) is being used, or their secondary item (which is not supported). In other words, for Romance 1.1, 1.2, or 2.0, this must always be the number 1.

In the first form, the user wants to use their equipment on an arbitrary point in space, whose coördinates are passed in.

In the second form, the user wants to use their equipment on a particular item or character. The optional `of` helps narrow down whether it should be an item or character.

WRITEME

### 8.741.3 Changes from 1.2 to 2.0

WRITEME

### 8.741.4 File

Defined in file `src/infinity/legacy-commands.lisp`.

## 8.742 Tootsville::Infinity-Wardrobe

### 8.742.1 Function

Infinity-Wardrobe names a function, with lambda list (D U RECIPIENT/S):

Describe what your Toot is wearing.

Lisp `WARDROBE = JSON wardrobe`

Note that several other commands will actually return wardrobe information packets.

### 8.742.2 Usage

This command requires no parameters.

### 8.742.3 200 OK

The returned packet, aside from the expected `status: true, from: "wardrobe"`, contains a key `wardrobe` which in turn contains a key `avatar` which itself contains the JSON data in the format of Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534, q.v.

```
{ from: "wardrobe",  
  status: true,  
  wardrobe: { avatar: { AVATAR INFO } } }
```

### 8.742.4 Changes from 1.2 to 2.0

The actual Section 8.742 [TOOTSVILLE INFINITY-WARDROBE], page 1026, function is new, but the returned packets `from: "wardrobe"` were already being used by other commands, including Section 8.687 [TOOTSVILLE INFINITY-DON], page 946, and Section 8.685 [TOOTSVILLE INFINITY-DOFF], page 944, and Section 8.686 [TOOTSVILLE INFINITY-DOFFF], page 945.

### 8.742.5 File

Defined in file `src/infinity/new-commands-20.lisp`.



## 8.743 Tootsville::Infinity-Websocket-Resource

### 8.743.1 Class

Infinity-Websocket-Resource names a class, with one superclass: HUNCHENSOCKET::WEBSOCKET-RESOURCE (not in this manual).

### 8.743.2 Slots

Class Infinity-Websocket-Resource has 3 direct slot definitions:

**Clients** (undocumented)

**Client-Class**  
(undocumented)

**Lock** (undocumented)

## 8.744 Tootsville::Infinity-Wtl

### 8.744.1 Function

Infinity-Wtl names a function, with lambda list (D U R):

Walk the Line

Lisp WTL = JSON wtl

Users send a “wtl” packet when they’re moving in a straight line; while other (arc) shapes were considered, they’re not currently supported. Each “wtl” packet has a start and end point, a start time, and a speed; this course is enough information for other clients to determine where along the line (linear interpolation) the walker is now.

### 8.744.2 Usage

```
{ course:
  { startPosition: { x: y: z: },
    startTime: UNIX-TIME,
    endPosition: { x: y: z: },
    speed: SPEED },
  facing: RADIANS }
```

Facing can be given as Section 8.750 [TOOTSVILLE INTERPRET-FACING], page 1035, and will be converted to decimal radians.

In return, all observers receive these “wtl” packets back ... WRITEME

### 8.744.3 Reply

```
{ from: "wtl", status: true, course: {}, facing:, u: UUID, n: NAME }
```

### 8.744.4 Future Directions

There is limited, partial, and broken support for the new d3 system which will eventually supersede wtl in Romance 2, but it is not useful today. Use wtl in new code.

### 8.744.5 See Also

See Section 8.733 [TOOTSVILLE INFINITY-SET-USER-VAR], page 1012, for discussion of an alternative way to submit a wtl packet or the legacy d form (see below).

### 8.744.6 Changes from 1.1

In Romance 1.0 and 1.1, the usual way to walk was using the d user variable, which was a string encoding very similar to the intent of wtl. The d string was developed by Robert Dawson and Bruce-Robert Pocock to cover up network latency by providing a concrete (linear interpolated) position for each character at all times, no matter how laggy players’ network connections were.

In the Persephone client software, these were sometimes referred to as a datList.

d strings consisted of a ~ delimited list of either x1~z1~x2~z2~facing~startTime or x1~y1~x2~y2~facing~startTime~z1~z2. Note how the two-coördinate form uses x,z with y pinned at zero.

As with wtl, facing can be supplied in either radians or as a value from the list N NE E SE S SW W NW. See Section 8.750 [TOOTSVILLE INTERPRET-FACING], page 1035.

### **8.744.7 File**

Defined in file `src/infinity/new-commands-20.lisp`.

## 8.745 Tootsville::Infinity-Wtl-4

### 8.745.1 Function

Infinity-Wtl-4 names a function, with lambda list (D U R):

Walk the Line indirect refresher from observer

Lisp WTL-4 = JSON wtl4

### 8.745.2 Usage

```
{ u: "TOOT-NAME",  
  course: { COURSE },  
  facing: RADIANS }
```

Facing can be provided as per Section 8.750 [TOOTSVILLE INTERPRET-FACING], page 1035.

WRITEME

### 8.745.3 File

Defined in file src/infinity/new-commands-20.lisp.

## **8.746 Tootsville::Init-Async**

### **8.746.1 Function**

Init-Async names a function, with lambda list NIL:

Initialize LPARALLEL for running tasks asynchronously.

### **8.746.2 File**

Defined in file src/main.lisp.

## **8.747 Tootsville::Init-Characters**

### **8.747.1 Function**

Init-Characters names a function, with lambda list NIL:

Initialize non-player characters in the game world.

### **8.747.2 File**

Defined in file src/characters/characters.lisp.

## 8.748 Tootsville::Integer-To-Byte-Vector

### 8.748.1 Function

Integer-To-Byte-Vector names a function, with lambda list (INTEGER &OPTIONAL (VECTOR (MAKE-ARRAY (CEILING (INTEGER-LENGTH INTEGER) 8) ELEMENT-TYPE (QUOTE (UNSIGNED-BYTE 8))))):

Convert INTEGER into VECTOR of (UNSIGNED-BYTE 8)

If VECTOR is supplied, it must be long enough to accept INTEGER without growing. Otherwise, the vector of the minimum length to hold INTEGER will be constructed.

The byte vector will be in big-endian (aka “network”) byte order.

### 8.748.2 File

Defined in file src/types/binary.lisp.

## 8.749 Tootsville::Integer-To-Color24

### 8.749.1 Function

Integer-To-Color24 names a function, with lambda list (NUMBER):

Return a color represented by the 24-bit integer NUMBER.

The upper 8 bits are the red channel; the next 8 bits, green; and the lowest 8 bits, the blue channel.

### 8.749.2 File

Defined in file `src/types/color+pattern.lisp`.



## 8.750 Tootsville::Interpret-Facing

### 8.750.1 Function

Interpret-Facing names a function, with lambda list (FACING):

Given a FACING string, return an angle in radians.

This supports a string that is a floating-point number of radians that can be parsed by `ORG.MAPCAR.PARSE-NUMBER::PARSE-NUMBER` (not in this manual) or one of the cardinal eight directions as a string: N NE E SE S SW W NW.

### 8.750.2 Changes from 1.2 to 2.0

Facing directions used to be *only* the cardinal directions; now, an arbitrary rotation in radians is possible.

TODO: Throw a 400-type exception when junk is passed in.

### 8.750.3 File

Defined in file `src/infinity/new-commands-20.lisp`.

## 8.751 Tootsville::Invalidate-Cache

### 8.751.1 Function

Invalidate-Cache names a function, with lambda list (OBJECT):

Identify that the cache is dirty and should be cleared of a certain set of possible records.

This is called by the :AFTER methods of Section 8.1115 [TOOTSVILLE SAVE-RECORD], page 1400, and Section 8.367 [TOOTSVILLE DESTROY-RECORD], page 623.

### 8.751.2 File

Defined in file src/db/generic-db.lisp.

## 8.752 Tootsville::Inventory-Item

### 8.752.1 Class

Inventory-Item names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.752.2 Slots

Class Inventory-Item has 4 direct slot definitions:

Item

Person

Toot

Equipped

## **8.753 Tootsville::Inventory-Item-Equipped**

### **8.753.1 Function**

Inventory-Item-Equipped names an undocumented function, with lambda list (INSTANCE).

### **8.753.2 File**

Defined in file src/db/friendly.lisp.

### **8.753.3 SetF Function**

(SETF Inventory-Item-Equipped) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.753.4 File**

Defined in file src/db/friendly.lisp.

## **8.754 Tootsville::Inventory-Item-Equipped-P**

### **8.754.1 Function**

Inventory-Item-Equipped-P names a function, with lambda list (ITEM):

Is the inventory item equipped at all?

### **8.754.2 File**

Defined in file src/toots.lisp.

## **8.755 Tootsville::Inventory-Item-Item**

### **8.755.1 Function**

Inventory-Item-Item names an undocumented function, with lambda list (INSTANCE).

### **8.755.2 File**

Defined in file src/db/friendly.lisp.

### **8.755.3 SetF Function**

(SETF Inventory-Item-Item) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.755.4 File**

Defined in file src/db/friendly.lisp.

## **8.756 Tootsville::Inventory-Item-P**

### **8.756.1 Function**

Inventory-Item-P names an undocumented function, with lambda list (OBJECT).

### **8.756.2 File**

Defined in file src/db/friendly.lisp.

## **8.757 Tootsville::Inventory-Item-Person**

### **8.757.1 Function**

Inventory-Item-Person names an undocumented function, with lambda list (INSTANCE).

### **8.757.2 File**

Defined in file src/db/friendly.lisp.

### **8.757.3 SetF Function**

(SETF Inventory-Item-Person) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.757.4 File**

Defined in file src/db/friendly.lisp.



## **8.758 Tootsville::Inventory-Item-Toot**

### **8.758.1 Function**

Inventory-Item-Toot names an undocumented function, with lambda list (INSTANCE).

### **8.758.2 File**

Defined in file src/db/friendly.lisp.

### **8.758.3 SetF Function**

(SETF Inventory-Item-Toot) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.758.4 File**

Defined in file src/db/friendly.lisp.

## 8.759 Tootsville::Item

### 8.759.1 Class

Item names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.759.2 Slots

Class Item has 16 direct slot definitions:

Uuid

Base-Color

Alt-Color

Template

Energy

Avatar-Scale-X

Avatar-Scale-Y

Avatar-Scale-Z

X

Y

Z

Facing

Latitude

Longitude

Altitude

World

## **8.760 Tootsville::Item-Alt-Color**

### **8.760.1 Function**

Item-Alt-Color names an undocumented function, with lambda list (INSTANCE).

### **8.760.2 File**

Defined in file src/db/friendly.lisp.

### **8.760.3 SetF Function**

(SETF Item-Alt-Color) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.760.4 File**

Defined in file src/db/friendly.lisp.

## **8.761 Tootsville::Item-Altitude**

### **8.761.1 Function**

Item-Altitude names an undocumented function, with lambda list (INSTANCE).

### **8.761.2 File**

Defined in file src/db/friendly.lisp.

### **8.761.3 SetF Function**

(SETF Item-Altitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.761.4 File**

Defined in file src/db/friendly.lisp.

## **8.762 Tootsville::Item-Avatar-Scale-X**

### **8.762.1 Function**

Item-Avatar-Scale-X names an undocumented function, with lambda list (INSTANCE).

### **8.762.2 File**

Defined in file src/db/friendly.lisp.

### **8.762.3 SetF Function**

(SETF Item-Avatar-Scale-X) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.762.4 File**

Defined in file src/db/friendly.lisp.

## **8.763 Tootsville::Item-Avatar-Scale-Y**

### **8.763.1 Function**

Item-Avatar-Scale-Y names an undocumented function, with lambda list (INSTANCE).

### **8.763.2 File**

Defined in file src/db/friendly.lisp.

### **8.763.3 SetF Function**

(SETF Item-Avatar-Scale-Y) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.763.4 File**

Defined in file src/db/friendly.lisp.

## 8.764 Tootsville::Item-Avatar-Scale-Z

### 8.764.1 Function

Item-Avatar-Scale-Z names an undocumented function, with lambda list (INSTANCE).

### 8.764.2 File

Defined in file src/db/friendly.lisp.

### 8.764.3 SetF Function

(SETF Item-Avatar-Scale-Z) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.764.4 File

Defined in file src/db/friendly.lisp.

## **8.765 Tootsville::Item-Base-Color**

### **8.765.1 Function**

Item-Base-Color names an undocumented function, with lambda list (INSTANCE).

### **8.765.2 File**

Defined in file src/db/friendly.lisp.

### **8.765.3 SetF Function**

(SETF Item-Base-Color) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.765.4 File**

Defined in file src/db/friendly.lisp.



## **8.766 Tootsville::Item-Energy**

### **8.766.1 Function**

Item-Energy names an undocumented function, with lambda list (INSTANCE).

### **8.766.2 File**

Defined in file src/db/friendly.lisp.

### **8.766.3 SetF Function**

(SETF Item-Energy) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.766.4 File**

Defined in file src/db/friendly.lisp.

## **8.767 Tootsville::Item-Facing**

### **8.767.1 Function**

Item-Facing names an undocumented function, with lambda list (INSTANCE).

### **8.767.2 File**

Defined in file src/db/friendly.lisp.

### **8.767.3 SetF Function**

(SETF Item-Facing) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.767.4 File**

Defined in file src/db/friendly.lisp.

## 8.768 Tootsville::Item-Gain-Energy

### 8.768.1 Function

Item-Gain-Energy names a function, with lambda list (ITEM AMOUNT):

Increase the energy of ITEM by AMOUNT (stopping at its :Energy-Max).

If ITEM's Energy-Kind is :COUNTABLE, then AMOUNT must be an integer.

### 8.768.2 File

Defined in file src/items.lisp.

## **8.769 Tootsville::Item-In-Inventory-P**

### **8.769.1 Function**

Item-In-Inventory-P names a function, with lambda list (ITEM):

Is ITEM in a character's inventory?

### **8.769.2 File**

Defined in file src/world.lisp.

## 8.770 Tootsville::Item-Info

### 8.770.1 Function

Item-Info names a function, with lambda list (ITEM):

Describes ITEM in a JSON structure.

This structure has the following keys:

<code>uuid</code>	The universally-unique ID of this particular item
<code>baseColor</code>	The base color of this item (if any). This is a primary color texture that may be applied to the model. See Section 8.945 [TOOTSVILLE PARSE-COLOR24], page 1230, for the syntax.
<code>altColor</code>	The alternate color of this item (if any). This is a secondary color texture that may be applied to the model. See Section 8.945 [TOOTSVILLE PARSE-COLOR24], page 1230, for the syntax.
<code>template</code>	The Item Template of which this individual item is an instance. This is a table in the form described at Section 8.787 [TOOTSVILLE ITEM-TEMPLATE-INFO], page 1072.
<code>energy</code>	For items with an <code>energyKind</code> of <code>COUNTABLE</code> or <code>UNCOUNTABLE</code> , this indicates the number of ( <code>COUNTABLE</code> ) discrete energy units remaining or ( <code>UNCOUNTABLE</code> ) the portion of <code>energyMax</code> remaining (which should be surfaced to the user as a percentage or the like).
<code>scape</code>	The item's scaling factors in each of the <code>x</code> , <code>y</code> , and <code>z</code> dimensions, as compared to the size of the raw asset in the avatar model file.
<code>position</code>	The item's relative position in <code>x</code> , <code>y</code> , and <code>z</code> coordinates
<code>facing</code>	The angle in which the item is facing in radians. Clients should also support, for compatibility, the eight cardinal directions given as the strings <code>N</code> , <code>NE</code> , <code>E</code> , <code>SE</code> , <code>S</code> , <code>SW</code> , <code>W</code> , or <code>NW</code> .
<code>world</code>	The world in which the item is located
<code>location</code>	The location of the item within the world in <code>lat-itude</code> , <code>long-itude</code> , and <code>alt-itude</code> .

### 8.770.2 File

Defined in file `src/items.lisp`.

## **8.771 Tootsville::Item-Latitude**

### **8.771.1 Function**

Item-Latitude names an undocumented function, with lambda list (INSTANCE).

### **8.771.2 File**

Defined in file src/db/friendly.lisp.

### **8.771.3 SetF Function**

(SETF Item-Latitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.771.4 File**

Defined in file src/db/friendly.lisp.

## 8.772 Tootsville::Item-Longitude

### 8.772.1 Function

Item-Longitude names an undocumented function, with lambda list (INSTANCE).

### 8.772.2 File

Defined in file src/db/friendly.lisp.

### 8.772.3 SetF Function

(SETF Item-Longitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.772.4 File

Defined in file src/db/friendly.lisp.

## **8.773 Tootsville::Item-Lose-Energy**

### **8.773.1 Function**

Item-Lose-Energy names a function, with lambda list (ITEM AMOUNT):

Decrease the energy of ITEM by AMOUNT (stopping at zero).

If the item's energy reaches zero, the effect of its :On-Zero flag will occur; either it will remain :EMPTY, or :VANISH.

If ITEM's Energy-Kind is :COUNTABLE, then AMOUNT must be an integer.

### **8.773.2 File**

Defined in file src/items.lisp.



## 8.774 Tootsville::Item-Owned-By-P

### 8.774.1 Function

Item-Owned-By-P names a function, with lambda list (ITEM &OPTIONAL (TOOT \*TOOT\*)):

A generalized boolean indicating whether ITEM is owned by TOOT.

Calls Section 8.1250 [TOOTSVILLE TOOT-INVENTORY], page 1537, to benefit from caching.

### 8.774.2 File

Defined in file src/items.lisp.

## **8.775 Tootsville::Item-P**

### **8.775.1 Function**

Item-P names an undocumented function, with lambda list (OBJECT).

### **8.775.2 File**

Defined in file src/db/friendly.lisp.

## 8.776 Tootsville::Item-Template

### 8.776.1 Function

Item-Template names an undocumented function, with lambda list (INSTANCE).

### 8.776.2 File

Defined in file src/db/friendly.lisp.

### 8.776.3 SetF Function

(SETF Item-Template) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.776.4 File

Defined in file src/db/friendly.lisp.

### 8.776.5 Class

Item-Template names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.776.6 Slots

Class Item-Template has 15 direct slot definitions:

Id

Name

Description

Trade

Default-Base-Color

Default-Alt-Color

Avatar

Energy-Kind

Energy-Max

On-Zero

Wear-Slot

Weight

Avatar-Scale-X

Avatar-Scale-Y

Avatar-Scale-Z

## **8.777 Tootsville::Item-Template-Avatar**

### **8.777.1 Function**

Item-Template-Avatar names an undocumented function, with lambda list (INSTANCE).

### **8.777.2 File**

Defined in file src/db/friendly.lisp.

### **8.777.3 SetF Function**

(SETF Item-Template-Avatar) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.777.4 File**

Defined in file src/db/friendly.lisp.

## 8.778 Tootsville::Item-Template-Avatar-Scale-X

### 8.778.1 Function

Item-Template-Avatar-Scale-X names an undocumented function, with lambda list (INSTANCE).

### 8.778.2 File

Defined in file src/db/friendly.lisp.

### 8.778.3 SetF Function

(SETF Item-Template-Avatar-Scale-X) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.778.4 File

Defined in file src/db/friendly.lisp.

## **8.779 Tootsville::Item-Template-Avatar-Scale-Y**

### **8.779.1 Function**

Item-Template-Avatar-Scale-Y names an undocumented function, with lambda list (INSTANCE).

### **8.779.2 File**

Defined in file src/db/friendly.lisp.

### **8.779.3 SetF Function**

(SETF Item-Template-Avatar-Scale-Y) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.779.4 File**

Defined in file src/db/friendly.lisp.

## 8.780 Tootsville::Item-Template-Avatar-Scale-Z

### 8.780.1 Function

Item-Template-Avatar-Scale-Z names an undocumented function, with lambda list (INSTANCE).

### 8.780.2 File

Defined in file src/db/friendly.lisp.

### 8.780.3 SetF Function

(SETF Item-Template-Avatar-Scale-Z) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.780.4 File

Defined in file src/db/friendly.lisp.

## **8.781 Tootsville::Item-Template-Default-Alt-Color**

### **8.781.1 Function**

Item-Template-Default-Alt-Color names an undocumented function, with lambda list (INSTANCE).

### **8.781.2 File**

Defined in file src/db/friendly.lisp.

### **8.781.3 SetF Function**

(SETF Item-Template-Default-Alt-Color) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.781.4 File**

Defined in file src/db/friendly.lisp.



## 8.782 Tootsville::Item-Template-Default-Base-Color

### 8.782.1 Function

Item-Template-Default-Base-Color names an undocumented function, with lambda list (INSTANCE).

### 8.782.2 File

Defined in file src/db/friendly.lisp.

### 8.782.3 SetF Function

(SETF Item-Template-Default-Base-Color) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.782.4 File

Defined in file src/db/friendly.lisp.

## **8.783 Tootsville::Item-Template-Description**

### **8.783.1 Function**

Item-Template-Description names an undocumented function, with lambda list (INSTANCE).

### **8.783.2 File**

Defined in file src/db/friendly.lisp.

### **8.783.3 SetF Function**

(SETF Item-Template-Description) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.783.4 File**

Defined in file src/db/friendly.lisp.

## 8.784 Tootsville::Item-Template-Energy-Kind

### 8.784.1 Function

Item-Template-Energy-Kind names an undocumented function, with lambda list (INSTANCE).

### 8.784.2 File

Defined in file src/db/friendly.lisp.

### 8.784.3 SetF Function

(SETF Item-Template-Energy-Kind) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.784.4 File

Defined in file src/db/friendly.lisp.

## **8.785 Tootsville::Item-Template-Energy-Max**

### **8.785.1 Function**

Item-Template-Energy-Max names an undocumented function, with lambda list (INSTANCE).

### **8.785.2 File**

Defined in file src/db/friendly.lisp.

### **8.785.3 SetF Function**

(SETF Item-Template-Energy-Max) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.785.4 File**

Defined in file src/db/friendly.lisp.

## **8.786 Tootsville::Item-Template-Id**

### **8.786.1 Function**

Item-Template-Id names an undocumented function, with lambda list (INSTANCE).

### **8.786.2 File**

Defined in file src/db/friendly.lisp.

### **8.786.3 SetF Function**

(SETF Item-Template-Id) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.786.4 File**

Defined in file src/db/friendly.lisp.

## 8.787 Tootsville::Item-Template-Info

### 8.787.1 Function

Item-Template-Info names a function, with lambda list (TEMPLATE):

Provides a JSON structure describing the item TEMPLATE given.

This structure has the following keys:

<code>id</code>	The unique ID (integer) of this item template						
<code>name</code>	The unique name of this item template. This may be user-visible.						
<code>description</code>	A description, which may be surfaced to the user, of this item.						
<code>trade</code>	This is one of the following values (case-insensitive): <table> <tr> <td><code>Y</code></td> <td>Yes, this item can be traded (given away or dropped).</td> </tr> <tr> <td><code>N</code></td> <td>No, this item cannot be traded (given away or dropped).</td> </tr> <tr> <td><code>X</code></td> <td>As <code>N</code>, but also, this item <i>should not</i> be visible to the player holding it (e.g. in inventory). This code represents the value “hidden”.</td> </tr> </table>	<code>Y</code>	Yes, this item can be traded (given away or dropped).	<code>N</code>	No, this item cannot be traded (given away or dropped).	<code>X</code>	As <code>N</code> , but also, this item <i>should not</i> be visible to the player holding it (e.g. in inventory). This code represents the value “hidden”.
<code>Y</code>	Yes, this item can be traded (given away or dropped).						
<code>N</code>	No, this item cannot be traded (given away or dropped).						
<code>X</code>	As <code>N</code> , but also, this item <i>should not</i> be visible to the player holding it (e.g. in inventory). This code represents the value “hidden”.						
<code>avatar</code>	The item avatar (model) representing this item.						
<code>energyKind</code>	The kind of energy (if any) used by this item. Values may be “null,” if the item does not consume any particular kind of energy, or <code>COUNTABLE</code> if the item uses a form of energy that is counted in discrete units, or <code>UNCOUNTABLE</code> if the item’s energy is recorded as a fluid percentage of its maximum value. This should be used by the client to provide either a counter, a gauge (meter), or no affordance indicating the energy level, as appropriate.						
<code>energyMax</code>	The maximum amount of energy that this kind of item can possess.						
<code>onZero</code>	When this item’s energy reaches zero, does it <code>VANISH</code> from the game, or remain <code>EMPTY</code> awaiting a refill?						
<code>gauge</code>	Linked to <code>energyKind</code> , should a gauge or counter be displayed? True for <code>EnergyKind</code> of <code>COUNTABLE</code> or <code>UNCOUNTABLE</code> , false for null.						
<code>wearSlot</code>	The ID of the wearable item slot into which this item can be equipped, if any.						
<code>weight</code>	The weight of instances of this template						

### 8.787.2 File

Defined in file `src/items.lisp`.

## **8.788 Tootsville::Item-Template-Name**

### **8.788.1 Function**

Item-Template-Name names an undocumented function, with lambda list (INSTANCE).

### **8.788.2 File**

Defined in file src/db/friendly.lisp.

### **8.788.3 SetF Function**

(SETF Item-Template-Name) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.788.4 File**

Defined in file src/db/friendly.lisp.

## **8.789 Tootsville::Item-Template-On-Zero**

### **8.789.1 Function**

Item-Template-On-Zero names an undocumented function, with lambda list (INSTANCE).

### **8.789.2 File**

Defined in file src/db/friendly.lisp.

### **8.789.3 SetF Function**

(SETF Item-Template-On-Zero) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.789.4 File**

Defined in file src/db/friendly.lisp.



## **8.790 Tootsville::Item-Template-P**

### **8.790.1 Function**

Item-Template-P names an undocumented function, with lambda list (OBJECT).

### **8.790.2 File**

Defined in file src/db/friendly.lisp.

## **8.791 Tootsville::Item-Template-Trade**

### **8.791.1 Function**

Item-Template-Trade names an undocumented function, with lambda list (INSTANCE).

### **8.791.2 File**

Defined in file src/db/friendly.lisp.

### **8.791.3 SetF Function**

(SETF Item-Template-Trade) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.791.4 File**

Defined in file src/db/friendly.lisp.

## 8.792 Tootsville::Item-Template-Wear-Slot

### 8.792.1 Function

Item-Template-Wear-Slot names an undocumented function, with lambda list (INSTANCE).

### 8.792.2 File

Defined in file src/db/friendly.lisp.

### 8.792.3 SetF Function

(SETF Item-Template-Wear-Slot) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.792.4 File

Defined in file src/db/friendly.lisp.

## **8.793 Tootsville::Item-Template-Weight**

### **8.793.1 Function**

Item-Template-Weight names an undocumented function, with lambda list (INSTANCE).

### **8.793.2 File**

Defined in file src/db/friendly.lisp.

### **8.793.3 SetF Function**

(SETF Item-Template-Weight) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.793.4 File**

Defined in file src/db/friendly.lisp.

## 8.794 Tootsville::Item-Uuid

### 8.794.1 Function

Item-Uuid names an undocumented function, with lambda list (INSTANCE).

### 8.794.2 File

Defined in file src/db/friendly.lisp.

### 8.794.3 SetF Function

(SETF Item-Uuid) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.794.4 File

Defined in file src/db/friendly.lisp.

## **8.795 Tootsville::Item-World**

### **8.795.1 Function**

Item-World names an undocumented function, with lambda list (INSTANCE).

### **8.795.2 File**

Defined in file src/db/friendly.lisp.

### **8.795.3 SetF Function**

(SETF Item-World) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.795.4 File**

Defined in file src/db/friendly.lisp.

## **8.796 Tootsville::Item-X**

### **8.796.1 Function**

Item-X names an undocumented function, with lambda list (INSTANCE).

### **8.796.2 File**

Defined in file src/db/friendly.lisp.

### **8.796.3 SetF Function**

(SETF Item-X) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.796.4 File**

Defined in file src/db/friendly.lisp.

## **8.797 Tootsville::Item-Y**

### **8.797.1 Function**

Item-Y names an undocumented function, with lambda list (INSTANCE).

### **8.797.2 File**

Defined in file src/db/friendly.lisp.

### **8.797.3 SetF Function**

(SETF Item-Y) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.797.4 File**

Defined in file src/db/friendly.lisp.



## **8.798 Tootsville::Item-Z**

### **8.798.1 Function**

Item-Z names an undocumented function, with lambda list (INSTANCE).

### **8.798.2 File**

Defined in file src/db/friendly.lisp.

### **8.798.3 SetF Function**

(SETF Item-Z) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.798.4 File**

Defined in file src/db/friendly.lisp.

## **8.799 Tootsville::Items-At**

### **8.799.1 Function**

Items-At names a function, with lambda list (LATITUDE LONGITUDE ALTITUDE WORLD):

All items in the space at LATITUDE, LONGITUDE, and ALTITUDE in WORLD.

Returns all items in that volume which are not in a character's inventory.

### **8.799.2 File**

Defined in file src/world.lisp.

## **8.800 Tootsville::Jack-Personality**

### **8.800.1 Class**

Jack-Personality names a class, with one superclass: Section 8.1086 [TOOTSVILLE ROBOT-JACK], page 1371.

This class defines a character named Jack

### **8.800.2 Slots**

Class Jack-Personality has no direct slots defined.

## 8.801 Tootsville::Journal

### 8.801.1 Function

Journal names a function, with lambda list (&REST WORDS):

Add a staff journal entry or review last entries.

UNIMPLEMENTED in 2.0

See also ‘INFINITY-JOURNAL’ for an Infinity Mode command for the same purpose.

### 8.801.2 Usage

To read the last journal entry, use `#journal #last`. To read the one before that, use `#journal #last -1`, and for farther back, use lesser (more negative) values of REFERENCE.

To read a journal entry relative to a certain user, use `#ref`.

To post a new journal entry, simply enter it after `#journal`. You can associate it with one or more users with `#for`.

```
#journal #last [REFERENCE]
```

```
#journal #ref USER [REFERENCE]
```

```
#journal [#for USER[,...]] ENTRY...
```

REFERENCE will always be zero or negative.

### 8.801.3 Examples

```
#journal #last
```

```
#journal #last -1
```

```
#journal #last -2
```

```
#journal #ref Pil
```

```
#journal #ref Pil -1
```

```
#journal #for pil,zap Added a new game with Pil and Zap
```

```
#journal #for mayor-louis Had to kick off mayor-louis for sedition
```

```
#journal Server game2 shut down for maintenance
```

### 8.801.4 File

Defined in file `src/infinity/modern-ops.lisp`.

## 8.802 Tootsville::Json-To-Html

### 8.802.1 Function

Json-To-Html names a function, with lambda list (JSON):

Converts JSON to a set of key-value pairs in pretty-printed HTML form.

### 8.802.2 File

Defined in file src/infinity/legacy-ops.lisp.

## **8.803 Tootsville::Kick**

### **8.803.1 Function**

Kick names a function, with lambda list (CLIENT TITLE MESSAGE REASON-CODE):

Kick CLIENT off with TITLE and MESSAGE and REASON-CODE.

### **8.803.2 File**

Defined in file src/websockets.lisp.

## **8.804 Tootsville::Kick-Child-Time-Up**

### **8.804.1 Function**

Kick-Child-Time-Up names a function, with lambda list (TOOT):

Kick TOOT as the child's time to play has expired.

### **8.804.2 File**

Defined in file src/websockets.lisp.

## **8.805 Tootsville::Kind-Of-Habitat**

### **8.805.1 Type**

Kind-Of-Habitat names a TYPE:

The various kinds of habitat that exist in the world.

**Shaddow**

**Rocky**

**Swamp**

**Grassland**

**Desert**

**Savannah**

**Forest**

**Ocean**

**Ice**

**Moon**

**Pink Moon**

**Moon Base**

**City**

**Farm**

**Manatee City**

**Beachside**

**Space**

**Asteroid Field**



## 8.806 Tootsville::Lambda-List-As-Variables

### 8.806.1 Function

Lambda-List-As-Variables names a function, with lambda list (*A-LIST*):

Convert *A-LIST* into variables for an endpoint function.

### 8.806.2 File

Defined in file src/web.lisp.

## **8.807 Tootsville::Last-Active**

### **8.807.1 Function**

Last-Active names an undocumented function, with lambda list (OBJECT).

### **8.807.2 SetF Function**

(SETF Last-Active) names an undocumented function, with lambda list (NEW-VALUE OBJECT).

## **8.808 Tootsville::Latitude**

### **8.808.1 Function**

Latitude names a function, with lambda list (THING):

The latitude of THING

### **8.808.2 File**

Defined in file src/characters/robots.lisp.

## 8.809 Tootsville::Legal-Age

### 8.809.1 Function

Legal-Age names a function, with lambda list (DATE-OF-BIRTH &OPTIONAL (REFERENCE-DATE (NOW))):

The age of a person born on DATE-OF-BIRTH, as of REFERENCE-DATE (or right now)

This uses the legal definition that the person's age increments at the midnight of their date of birth each year, with the date 29 February treated as 1 March on non-leap-years.

The time zone used for this computation is the not defined, however, yielding rather irregular behaviour depending on time zones and the like.

TODO: Determine in what time zone we should compute this for legal reasons, eg, COPPA.

### 8.809.2 File

Defined in file src/types/date+time.lisp.

## **8.810 Tootsville::Lil-Mc-Personality**

### **8.810.1 Class**

Lil-Mc-Personality names a class, with one superclass: Section 8.1087 [TOOTSVILLE ROBOT-LIL-MC], page 1372.

This class defines a character named Lil-Mc

### **8.810.2 Slots**

Class Lil-Mc-Personality has no direct slots defined.

## 8.811 Tootsville::Limit-String-Length

### 8.811.1 Function

Limit-String-Length names a function, with lambda list (STRING LENGTH):

Returns up to LENGTH characters from STRING.

If STRING is less than LENGTH characters in length, the entire (original) string is returned.

### 8.811.2 File

Defined in file src/types/string-characteristics.lisp.

## 8.812 Tootsville::Lisp-To-Db-Name

### 8.812.1 Function

Lisp-To-Db-Name names a function, with lambda list (NAME):

Convert a Lispy name to an SQL-type one.

Particularly, changes CAPS-WITH-KEBABS to lower\_with\_snakes.

### 8.812.2 File

Defined in file src/db/db-central.lisp.

## **8.813 Tootsville::List-Banhammers**

### **8.813.1 Function**

List-Banhammers names an undocumented function, with lambda list NIL.

### **8.813.2 File**

Defined in file `src/infinity/legacy-ops.lisp`.



## 8.814 Tootsville::List-Of-String=

### 8.814.1 Function

List-Of-String= names an undocumented function, with lambda list (A B).

### 8.814.2 File

Defined in file src/users.lisp.

## **8.815 Tootsville::Listen-For-Websockets**

### **8.815.1 Function**

Listen-For-Websockets names a function, with lambda list NIL:

Start listening for websocket connections.

### **8.815.2 File**

Defined in file src/websockets.lisp.

## **8.816 Tootsville::Listener-Name**

### **8.816.1 Function**

Listener-Name names an undocumented function, with lambda list (LISTENER).

### **8.816.2 File**

Defined in file `src/characters/robots.lisp`.

## 8.817 Tootsville::Load-Config

### 8.817.1 Function

Load-Config names a function, with lambda list (&OPTIONAL (CONFIG-FILE (DEFAULT-CONFIG-FILE))):

Load the configuration from CONFIG-FILE.

### 8.817.2 File

Defined in file src/config.lisp.

## 8.818 Tootsville::Load-Record

### 8.818.1 Function

Load-Record names a function, with lambda list (TYPE COLUMNS):

Create an object of TYPE from the raw data in COLUMNS.

Used by Section 8.575 [TOOTSVILLE FIND-RECORD], page 832, and Section 8.576 [TOOTSVILLE FIND-RECORDS], page 833, which are what a normal user of this API will be interested-in.

### 8.818.2 File

Defined in file src/db/generic-db.lisp.

## **8.819 Tootsville::Local-Room-Vars**

### **8.819.1 Function**

Local-Room-Vars names a function, with lambda list NIL:

Gets “room variables” local to \*CLIENT\*.

See Section 8.702 [TOOTSVILLE INFINITY-GET-ROOM-VARS], page 969, for a discussion.

### **8.819.2 File**

Defined in file src/infinity/new-commands-20.lisp.

## 8.820 Tootsville::Locale-Music

### 8.820.1 Class

Locale-Music names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.820.2 Slots

Class Locale-Music has 5 direct slot definitions:

Music

X

Y

Z

Radius

## **8.821 Tootsville::Locale-Music-Music**

### **8.821.1 Function**

Locale-Music-Music names an undocumented function, with lambda list (INSTANCE).

### **8.821.2 File**

Defined in file src/db/friendly.lisp.

### **8.821.3 SetF Function**

(SETF Locale-Music-Music) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.821.4 File**

Defined in file src/db/friendly.lisp.



## **8.822 Tootsville::Locale-Music-P**

### **8.822.1 Function**

Locale-Music-P names an undocumented function, with lambda list (OBJECT).

### **8.822.2 File**

Defined in file src/db/friendly.lisp.

## **8.823 Tootsville::Locale-Music-Radius**

### **8.823.1 Function**

Locale-Music-Radius names an undocumented function, with lambda list (INSTANCE).

### **8.823.2 File**

Defined in file src/db/friendly.lisp.

### **8.823.3 SetF Function**

(SETF Locale-Music-Radius) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.823.4 File**

Defined in file src/db/friendly.lisp.

## 8.824 Tootsville::Locale-Music-X

### 8.824.1 Function

Locale-Music-X names an undocumented function, with lambda list (INSTANCE).

### 8.824.2 File

Defined in file src/db/friendly.lisp.

### 8.824.3 SetF Function

(SETF Locale-Music-X) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.824.4 File

Defined in file src/db/friendly.lisp.

## **8.825 Tootsville::Locale-Music-Y**

### **8.825.1 Function**

Locale-Music-Y names an undocumented function, with lambda list (INSTANCE).

### **8.825.2 File**

Defined in file src/db/friendly.lisp.

### **8.825.3 SetF Function**

(SETF Locale-Music-Y) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.825.4 File**

Defined in file src/db/friendly.lisp.

## 8.826 Tootsville::Locale-Music-Z

### 8.826.1 Function

Locale-Music-Z names an undocumented function, with lambda list (INSTANCE).

### 8.826.2 File

Defined in file src/db/friendly.lisp.

### 8.826.3 SetF Function

(SETF Locale-Music-Z) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.826.4 File

Defined in file src/db/friendly.lisp.

## 8.827 Tootsville::Login

### 8.827.1 Class

Login names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.827.2 Slots

Class Login has 7 direct slot definitions:

Uuid

Person

Credential

Start

Renewed

Last-Seen

Origin

## 8.828 Tootsville::Login-Child

### 8.828.1 Function

Login-Child names a function, with lambda list (TOOT):

Start a login request for TOOT, if one is not already pending.

WRITEME

### 8.828.2 File

Defined in file src/users.lisp.

## **8.829 Tootsville::Login-Credential**

### **8.829.1 Function**

Login-Credential names an undocumented function, with lambda list (INSTANCE).

### **8.829.2 File**

Defined in file src/db/friendly.lisp.

### **8.829.3 SetF Function**

(SETF Login-Credential) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.829.4 File**

Defined in file src/db/friendly.lisp.



## 8.830 Tootsville::Login-Fail

### 8.830.1 Function

Login-Fail names a function, with lambda list (ERR2 MSG CLIENT):

Sends a login failure message.

```
{ from: "login",
  status: false,
  err: "login.fail",
  msg: "User-visible error message",
  err2: "unique error token" }
```

`err2` is an unique error token to identify the specific reason why login was denied in machine-readable form. `msg` conveys approximately the same information, but in a form suitable for relaying to the end user.

#### hashfail

The password hash submitted was incorrect. Refer to Section 8.713 [TOOTSVILLE INFINITY-LOGIN], page 983, for the proper structure of the login packet. This usually, if the software is conforming, means that the user entered a bad password.

#### no-apple

The client did not obtain an apple with which to create the password hash.

#### no-Toot

The Toot named does not exist.

#### no-zone

The Zone named does not exist. Only `$Eden` is a valid Zone name.

#### not-child

The Toot named was not a child Toot. Login by password is for children only; adults use third-party authentication (eg, Firebase for Google and Twitter) to log in.

For an overview of the child login process, see Section 8.693 [TOOTSVILLE INFINITY-GET-APPLE], page 956.

### 8.830.2 File

Defined in file `src/websockets.lisp`.

## **8.831 Tootsville::Login-Failed-Message**

### **8.831.1 Function**

Login-Failed-Message names a function, with lambda list NIL:

Produce a logOK for failed login

### **8.831.2 File**

Defined in file src/websockets.lisp.

## 8.832 Tootsville::Login-Last-Seen

### 8.832.1 Function

Login-Last-Seen names an undocumented function, with lambda list (INSTANCE).

### 8.832.2 File

Defined in file src/db/friendly.lisp.

### 8.832.3 SetF Function

(SETF Login-Last-Seen) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.832.4 File

Defined in file src/db/friendly.lisp.

## **8.833 Tootsville::Login-Ok-Message**

### **8.833.1 Function**

Login-Ok-Message names a function, with lambda list NIL:

Produce a logOK message for successful login

### **8.833.2 File**

Defined in file src/websockets.lisp.

## 8.834 Tootsville::Login-Origin

### 8.834.1 Function

Login-Origin names an undocumented function, with lambda list (INSTANCE).

### 8.834.2 File

Defined in file src/db/friendly.lisp.

### 8.834.3 SetF Function

(SETF Login-Origin) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.834.4 File

Defined in file src/db/friendly.lisp.

## **8.835 Tootsville::Login-P**

### **8.835.1 Function**

Login-P names an undocumented function, with lambda list (OBJECT).

### **8.835.2 File**

Defined in file src/db/friendly.lisp.

## 8.836 Tootsville::Login-Person

### 8.836.1 Function

Login-Person names an undocumented function, with lambda list (INSTANCE).

### 8.836.2 File

Defined in file src/db/friendly.lisp.

### 8.836.3 SetF Function

(SETF Login-Person) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.836.4 File

Defined in file src/db/friendly.lisp.

## **8.837 Tootsville::Login-Renewed**

### **8.837.1 Function**

Login-Renewed names an undocumented function, with lambda list (INSTANCE).

### **8.837.2 File**

Defined in file src/db/friendly.lisp.

### **8.837.3 SetF Function**

(SETF Login-Renewed) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.837.4 File**

Defined in file src/db/friendly.lisp.



## 8.838 Tootsville::Login-Start

### 8.838.1 Function

Login-Start names an undocumented function, with lambda list (INSTANCE).

### 8.838.2 File

Defined in file src/db/friendly.lisp.

### 8.838.3 SetF Function

(SETF Login-Start) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.838.4 File

Defined in file src/db/friendly.lisp.

## 8.839 Tootsville::Login-Uuid

### 8.839.1 Function

Login-Uuid names an undocumented function, with lambda list (INSTANCE).

### 8.839.2 File

Defined in file src/db/friendly.lisp.

### 8.839.3 SetF Function

(SETF Login-Uuid) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.839.4 File

Defined in file src/db/friendly.lisp.

## **8.840 Tootsville::Longitude**

### **8.840.1 Function**

Longitude names a function, with lambda list (THING):

The longitude of THING

### **8.840.2 File**

Defined in file src/characters/robots.lisp.

## **8.841 Tootsville::Look-For-Ssl-Certs**

### **8.841.1 Function**

Look-For-Ssl-Certs names an undocumented function, with lambda list NIL.

### **8.841.2 File**

Defined in file src/config.lisp.

## 8.842 Tootsville::Lot

### 8.842.1 Class

Lot names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.842.2 Slots

Class Lot has 9 direct slot definitions:

X1

X2

Y1

Y2

Z1

Z2

Ownership

Owner-Toot

World

## **8.843 Tootsville::Lot-Owner-Toot**

### **8.843.1 Function**

Lot-Owner-Toot names an undocumented function, with lambda list (INSTANCE).

### **8.843.2 File**

Defined in file src/db/friendly.lisp.

### **8.843.3 SetF Function**

(SETF Lot-Owner-Toot) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.843.4 File**

Defined in file src/db/friendly.lisp.

## 8.844 Tootsville::Lot-Ownership

### 8.844.1 Function

Lot-Ownership names an undocumented function, with lambda list (INSTANCE).

### 8.844.2 File

Defined in file src/db/friendly.lisp.

### 8.844.3 SetF Function

(SETF Lot-Ownership) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.844.4 File

Defined in file src/db/friendly.lisp.

## **8.845 Tootsville::Lot-P**

### **8.845.1 Function**

Lot-P names an undocumented function, with lambda list (OBJECT).

### **8.845.2 File**

Defined in file src/db/friendly.lisp.



## 8.846 Tootsville::Lot-World

### 8.846.1 Function

Lot-World names an undocumented function, with lambda list (INSTANCE).

### 8.846.2 File

Defined in file src/db/friendly.lisp.

### 8.846.3 SetF Function

(SETF Lot-World) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.846.4 File

Defined in file src/db/friendly.lisp.

## **8.847 Tootsville::Lot-X1**

### **8.847.1 Function**

Lot-X1 names an undocumented function, with lambda list (INSTANCE).

### **8.847.2 File**

Defined in file src/db/friendly.lisp.

### **8.847.3 SetF Function**

(SETF Lot-X1) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.847.4 File**

Defined in file src/db/friendly.lisp.

## 8.848 Tootsville::Lot-X2

### 8.848.1 Function

Lot-X2 names an undocumented function, with lambda list (INSTANCE).

### 8.848.2 File

Defined in file src/db/friendly.lisp.

### 8.848.3 SetF Function

(SETF Lot-X2) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.848.4 File

Defined in file src/db/friendly.lisp.

## **8.849 Tootsville::Lot-Y1**

### **8.849.1 Function**

Lot-Y1 names an undocumented function, with lambda list (INSTANCE).

### **8.849.2 File**

Defined in file src/db/friendly.lisp.

### **8.849.3 SetF Function**

(SETF Lot-Y1) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.849.4 File**

Defined in file src/db/friendly.lisp.

## 8.850 Tootsville::Lot-Y2

### 8.850.1 Function

Lot-Y2 names an undocumented function, with lambda list (INSTANCE).

### 8.850.2 File

Defined in file src/db/friendly.lisp.

### 8.850.3 SetF Function

(SETF Lot-Y2) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.850.4 File

Defined in file src/db/friendly.lisp.

## **8.851 Tootsville::Lot-Z1**

### **8.851.1 Function**

Lot-Z1 names an undocumented function, with lambda list (INSTANCE).

### **8.851.2 File**

Defined in file src/db/friendly.lisp.

### **8.851.3 SetF Function**

(SETF Lot-Z1) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.851.4 File**

Defined in file src/db/friendly.lisp.

## 8.852 Tootsville::Lot-Z2

### 8.852.1 Function

Lot-Z2 names an undocumented function, with lambda list (INSTANCE).

### 8.852.2 File

Defined in file src/db/friendly.lisp.

### 8.852.3 SetF Function

(SETF Lot-Z2) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.852.4 File

Defined in file src/db/friendly.lisp.

## 8.853 Tootsville::Make-Avatar

### 8.853.1 Function

Make-Avatar names an undocumented function, with lambda list (&KEY ((ID ID) NIL) ((MONIKER MONIKER) NIL) ((AVATAR-SCALE-X AVATAR-SCALE-X) NIL) ((AVATAR-SCALE-Y AVATAR-SCALE-Y) NIL) ((AVATAR-SCALE-Z AVATAR-SCALE-Z) NIL)).

### 8.853.2 File

Defined in file src/db/friendly.lisp.



## 8.854 Tootsville::Make-Avatar-Slot

### 8.854.1 Function

Make-Avatar-Slot names an undocumented function, with lambda list (&KEY ((ID ID) NIL) ((AVATAR AVATAR) NIL) ((SLOT SLOT) NIL) ((VALENCE VALENCE) NIL)).

### 8.854.2 File

Defined in file src/db/friendly.lisp.

## 8.855 Tootsville::Make-Character-Music

### 8.855.1 Function

Make-Character-Music names an undocumented function, with lambda list (&KEY ((MUSIC MUSIC) NIL) ((TOOT TOOT) NIL)).

### 8.855.2 File

Defined in file src/db/friendly.lisp.

## 8.856 Tootsville::Make-Child-Request

### 8.856.1 Function

Make-Child-Request names an undocumented function, with lambda list (&KEY ((UUID UUID) NIL) ((TOOT TOOT) NIL) ((PLACED-AT PLACED-AT) NIL) ((ALLOWED-AT ALLOWED-AT) NIL) ((DENIED-AT DENIED-AT) NIL) ((ALLOWED-FOR ALLOWED-FOR) NIL) ((RESPONSE RESPONSE) NIL)).

### 8.856.2 File

Defined in file src/db/friendly.lisp.

## 8.857 Tootsville::Make-Color24

### 8.857.1 Function

Make-Color24 names an undocumented function, with lambda list (&KEY ((RED RED) 0) ((GREEN GREEN) 0) ((BLUE BLUE) 0)).

### 8.857.2 File

Defined in file src/types/color+pattern.lisp.

## 8.858 Tootsville::Make-Contact

### 8.858.1 Function

Make-Contact names an undocumented function, with lambda list (&KEY ((UUID UUID) NIL) ((OWNER OWNER) NIL) ((CONTACT CONTACT) NIL) ((STARREDP STARREDP) NIL) ((ADDED ADDED) NIL) ((LAST-USED LAST-USED) NIL)).

### 8.858.2 File

Defined in file src/db/friendly.lisp.

## 8.859 Tootsville::Make-Credential

### 8.859.1 Function

Make-Credential names an undocumented function, with lambda list (&KEY ((UUID UUID) NIL) ((PERSON PERSON) NIL) ((UID UID) NIL) ((PROVIDER PROVIDER) NIL) ((ID-TOKEN ID-TOKEN) NIL) ((AUTH-TOKEN AUTH-TOKEN) NIL) ((REFRESH-TOKEN REFRESH-TOKEN) NIL) ((JSON-INFO JSON-INFO) NIL)).

### 8.859.2 File

Defined in file src/db/friendly.lisp.

## **8.860 Tootsville::Make-Endpoint-Function-Name**

### **8.860.1 Function**

Make-Endpoint-Function-Name names a function, with lambda list (METHOD URI ACCEPT-TYPE):

Create the name of the endpoint function for METHOD, URI, and ACCEPT-TYPE.

### **8.860.2 File**

Defined in file src/web.lisp.

## 8.861 Tootsville::Make-Game-Point

### 8.861.1 Function

Make-Game-Point names an undocumented function, with lambda list (&KEY ((LATITUDE LATITUDE) NIL) ((LONGITUDE LONGITUDE) NIL) ((ALTITUDE ALTITUDE) NIL) ((WORLD WORLD) NIL) ((X X) NIL) ((Y Y) NIL) ((Z Z) NIL)).

### 8.861.2 File

Defined in file src/characters/robots.lisp.



## 8.862 Tootsville::Make-Gossip-Initiation

### 8.862.1 Function

Make-Gossip-Initiation names an undocumented function, with lambda list (&KEY ((UUID UUID) NIL) ((ANSWER ANSWER) NIL)).

### 8.862.2 File

Defined in file src/gossip.lisp.

## **8.863 Tootsville::Make-Inventory-Item**

### **8.863.1 Function**

Make-Inventory-Item names an undocumented function, with lambda list (&KEY ((ITEM ITEM) NIL) ((PERSON PERSON) NIL) ((TOOT TOOT) NIL) ((EQUIPPED EQUIPPED) NIL)).

### **8.863.2 File**

Defined in file src/db/friendly.lisp.

## 8.864 Tootsville::Make-Item

### 8.864.1 Function

Make-Item names an undocumented function, with lambda list (&KEY ((UUID UUID) NIL) ((BASE-COLOR BASE-COLOR) NIL) ((ALT-COLOR ALT-COLOR) NIL) ((TEMPLATE TEMPLATE) NIL) ((ENERGY ENERGY) NIL) ((AVATAR-SCALE-X AVATAR-SCALE-X) NIL) ((AVATAR-SCALE-Y AVATAR-SCALE-Y) NIL) ((AVATAR-SCALE-Z AVATAR-SCALE-Z) NIL) ((X X) NIL) ((Y Y) NIL) ((Z Z) NIL) ((FACING FACING) NIL) ((LATITUDE LATITUDE) NIL) ((LONGITUDE LONGITUDE) NIL) ((ALTITUDE ALTITUDE) NIL) ((WORLD WORLD) NIL)).

### 8.864.2 File

Defined in file src/db/friendly.lisp.

## 8.865 Tootsville::Make-Item-Template

### 8.865.1 Function

Make-Item-Template names an undocumented function, with lambda list (&KEY ((ID ID) NIL) ((NAME NAME) NIL) ((DESCRIPTION DESCRIPTION) NIL) ((TRADE TRADE) NIL) ((DEFAULT-BASE-COLOR DEFAULT-BASE-COLOR) NIL) ((DEFAULT-ALT-COLOR DEFAULT-ALT-COLOR) NIL) ((AVATAR AVATAR) NIL) ((ENERGY-KIND ENERGY-KIND) NIL) ((ENERGY-MAX ENERGY-MAX) NIL) ((ON-ZERO ON-ZERO) NIL) ((WEAR-SLOT WEAR-SLOT) NIL) ((WEIGHT WEIGHT) NIL) ((AVATAR-SCALE-X AVATAR-SCALE-X) NIL) ((AVATAR-SCALE-Y AVATAR-SCALE-Y) NIL) ((AVATAR-SCALE-Z AVATAR-SCALE-Z) NIL)).

### 8.865.2 File

Defined in file src/db/friendly.lisp.

## 8.866 Tootsville::Make-Locale-Music

### 8.866.1 Function

Make-Locale-Music names an undocumented function, with lambda list (&KEY ((MUSIC MUSIC) NIL) ((X X) NIL) ((Y Y) NIL) ((Z Z) NIL) ((RADIUS RADIUS) NIL)).

### 8.866.2 File

Defined in file src/db/friendly.lisp.

## 8.867 Tootsville::Make-Login

### 8.867.1 Function

Make-Login names an undocumented function, with lambda list (&KEY ((UUID UUID) NIL) ((PERSON PERSON) NIL) ((CREDENTIAL CREDENTIAL) NIL) ((START START) NIL) ((RENEWED RENEWED) NIL) ((LAST-SEEN LAST-SEEN) NIL) ((ORIGIN ORIGIN) NIL)).

### 8.867.2 File

Defined in file src/db/friendly.lisp.

## 8.868 Tootsville::Make-Lot

### 8.868.1 Function

Make-Lot names an undocumented function, with lambda list (&KEY ((X1 X1) NIL) ((X2 X2) NIL) ((Y1 Y1) NIL) ((Y2 Y2) NIL) ((Z1 Z1) NIL) ((Z2 Z2) NIL) ((OWNER-SHIP OWNERSHIP) NIL) ((OWNER-TOOT OWNER-TOOT) NIL) ((WORLD WORLD) NIL)).

### 8.868.2 File

Defined in file src/db/friendly.lisp.

## 8.869 Tootsville::Make-Metronome-Task

### 8.869.1 Function

Make-Metronome-Task names an undocumented function, with lambda list (&KEY ((FREQUENCY FREQUENCY) NIL) ((ONE-SHOT-TIME ONE-SHOT-TIME) NIL) ((NAME NAME) NIL) ((FUNCTION FUNCTION) NIL) ((THREAD THREAD) NIL)).

### 8.869.2 File

Defined in file src/metronome.lisp.



## 8.870 Tootsville::Make-Mist

### 8.870.1 Function

Make-Mist names an undocumented function, with lambda list (&KEY ((WORLD WORLD) NIL) ((LATITUDE-1 LATITUDE-1) NIL) ((LONGITUDE-1 LONGITUDE-1) NIL) ((ALTITUDE-1 ALTITUDE-1) NIL) ((LATITUDE-2 LATITUDE-2) NIL) ((LONGITUDE-2 LONGITUDE-2) NIL) ((ALTITUDE-2 ALTITUDE-2) NIL) ((DEFINEDP DEFINEDP) NIL)).

### 8.870.2 File

Defined in file src/db/friendly.lisp.

## **8.871 Tootsville::Make-Music**

### **8.871.1 Function**

Make-Music names an undocumented function, with lambda list (&KEY ((ID ID) NIL) ((TITLE TITLE) NIL) ((ARTIST ARTIST) NIL) ((GENRE GENRE) NIL) ((LICENSE LICENSE) NIL) ((MONIKER MONIKER) NIL)).

### **8.871.2 File**

Defined in file src/db/friendly.lisp.

## 8.872 Tootsville::Make-New-Toot-State

### 8.872.1 Function

Make-New-Toot-State names a function, with lambda list (TOOT):

Set up the state for TOOT, who has never logged in before.

WRITEME

### 8.872.2 File

Defined in file `src/infinity/new-commands-20.lisp`.

## **8.873 Tootsville::Make-Parent-Child**

### **8.873.1 Function**

Make-Parent-Child names an undocumented function, with lambda list (&KEY ((PARENT PARENT) NIL) ((CHILD CHILD) NIL)).

### **8.873.2 File**

Defined in file src/db/friendly.lisp.

## 8.874 Tootsville::Make-Pattern

### 8.874.1 Function

Make-Pattern names an undocumented function, with lambda list (&KEY ((ID ID) NIL) ((NAME NAME) NIL)).

### 8.874.2 File

Defined in file src/db/friendly.lisp.

## 8.875 Tootsville::Make-Person

### 8.875.1 Function

Make-Person names an undocumented function, with lambda list (&KEY ((UUID UUID) NIL) ((DISPLAY-NAME DISPLAY-NAME) NIL) ((GIVEN-NAME GIVEN-NAME) NIL) ((SURNAME SURNAME) NIL) ((DATE-OF-BIRTH DATE-OF-BIRTH) NIL) ((AGE AGE) NIL) ((SENSITIVEP SENSITIVEP) NIL) ((GENDER GENDER) NIL) ((LANG LANG) NIL)).

### 8.875.2 File

Defined in file src/db/friendly.lisp.

## 8.876 Tootsville::Make-Person-Link

### 8.876.1 Function

Make-Person-Link names an undocumented function, with lambda list (&KEY ((UUID UUID) NIL) ((PERSON PERSON) NIL) ((REL REL) NIL) ((URL URL) NIL) ((LABEL LABEL) NIL) ((PROVENANCE PROVENANCE) NIL)).

### 8.876.2 File

Defined in file src/db/friendly.lisp.

## 8.877 Tootsville::Make-Place

### 8.877.1 Function

Make-Place names an undocumented function, with lambda list (&KEY ((UUID UUID) NIL) ((WORLD WORLD) NIL) ((LATITUDE LATITUDE) NIL) ((LONGITUDE LONGITUDE) NIL) ((ALTITUDE ALTITUDE) NIL) ((SHAPE SHAPE) NIL) ((KIND KIND) NIL) ((ATTRIBUTES ATTRIBUTES) NIL) ((APPEARANCE APPEARANCE) NIL)).

### 8.877.2 File

Defined in file src/db/friendly.lisp.



## 8.878 Tootsville::Make-Quaestor-Event

### 8.878.1 Function

Make-Quaestor-Event names an undocumented function, with lambda list (&KEY ((UUID UUID) NIL) ((SOURCE SOURCE) NIL) ((STARTED-BY STARTED-BY) NIL) ((STARTED-AT STARTED-AT) NIL) ((ENDED-AT ENDED-AT) NIL) ((COMPLETEDP COMPLETEDP) NIL) ((PEANUTS PEANUTS) NIL) ((FAIRY-DUST FAIRY-DUST) NIL) ((ITEM ITEM) NIL) ((SCORE SCORE) NIL) ((MEDAL MEDAL) NIL)).

### 8.878.2 File

Defined in file src/db/friendly.lisp.

## 8.879 Tootsville::Make-Record

### 8.879.1 Function

Make-Record names a function, with lambda list (TYPE &REST COLUMNS+VALUES):

Create a new record of TYPE with initial values COLUMNS+VALUES.

Implies saving that record to the backing storage, as well.

This is analogous to MAKE-INSTANCE (see the Common Lisp HyperSpec) or a DEFSTRUCT (see the Common Lisp HyperSpec) constructor, but for ORM objects.

### 8.879.2 File

Defined in file src/db/generic-db.lisp.

## 8.880 Tootsville::Make-Sms

### 8.880.1 Function

Make-Sms names an undocumented function, with lambda list (&KEY ((UUID UUID) NIL) ((SENDER SENDER) NIL) ((DESTINATION DESTINATION) NIL) ((MESSAGE MESSAGE) NIL) ((MMSP MMSP) NIL)).

### 8.880.2 File

Defined in file src/db/friendly.lisp.

## **8.881 Tootsville::Make-Store-Item**

### **8.881.1 Function**

Make-Store-Item names an undocumented function, with lambda list (&KEY ((UUID UUID) NIL) ((TEMPLATE TEMPLATE) NIL) ((QTY QTY) NIL) ((PRICE PRICE) NIL) ((CURRENCY CURRENCY) NIL)).

### **8.881.2 File**

Defined in file src/db/friendly.lisp.

## 8.882 Tootsville::Make-Tcp-Client

### 8.882.1 Function

Make-Tcp-Client names an undocumented function, with lambda list (&KEY ((SOCKET SOCKET) NIL) ((BUFFER BUFFER) NIL) ((EXPECTED-LENGTH EXPECTED-LENGTH) NIL) ((PEER PEER) NIL)).

### 8.882.2 File

Defined in file src/tcp-stream.lisp.

## 8.883 Tootsville::Make-Terrain-Height

### 8.883.1 Function

Make-Terrain-Height names an undocumented function, with lambda list (&KEY ((WORLD WORLD) NIL) ((LATITUDE LATITUDE) NIL) ((LONGITUDE LONGITUDE) NIL) ((TERRAIN TERRAIN) NIL)).

### 8.883.2 File

Defined in file src/db/friendly.lisp.

## 8.884 Tootsville::Make-Toot

### 8.884.1 Function

Make-Toot names an undocumented function, with lambda list (&KEY ((UUID UUID) NIL) ((NAME NAME) NIL) ((PATTERN PATTERN) NIL) ((BASE-COLOR BASE-COLOR) NIL) ((PATTERN-COLOR PATTERN-COLOR) NIL) ((PAD-COLOR PAD-COLOR) NIL) ((AVATAR AVATAR) NIL) ((PLAYER PLAYER) NIL) ((CHILD-CODE CHILD-CODE) NIL) ((LAST-ACTIVE LAST-ACTIVE) NIL) ((NOTE NOTE) NIL) ((AVATAR-SCALE-X AVATAR-SCALE-X) NIL) ((AVATAR-SCALE-Y AVATAR-SCALE-Y) NIL) ((AVATAR-SCALE-Z AVATAR-SCALE-Z) NIL)).

### 8.884.2 File

Defined in file src/db/friendly.lisp.

## 8.885 Tootsville::Make-Toot-Quiesced

### 8.885.1 Function

Make-Toot-Quiesced names an undocumented function, with lambda list (&KEY ((TOOT TOOT) NIL) ((WORLD WORLD) NIL) ((LATITUDE LATITUDE) NIL) ((LONGITUDE LONGITUDE) NIL) ((ALTITUDE ALTITUDE) NIL) ((WTL WTL) NIL) ((D3 D3) NIL) ((EMOTION EMOTION) NIL) ((OBSERVED OBSERVED) NIL) ((PEER-ADDRESS PEER-ADDRESS) NIL) ((ATTRIBS ATTRIBS) NIL)).

### 8.885.2 File

Defined in file src/db/friendly.lisp.



## 8.886 Tootsville::Make-Wear-Slot

### 8.886.1 Function

Make-Wear-Slot names an undocumented function, with lambda list (&KEY ((ID ID) NIL) ((NAME NAME) NIL) ((ALTERNATE ALTERNATE) NIL) ((AVATAR-POINT AVATAR-POINT) NIL) ((VALENCE VALENCE) NIL) ((OBSTRUCT-POINT OBSTRUCT-POINT) NIL) ((OBSTRUCT-MIN OBSTRUCT-MIN) NIL) ((OBSTRUCT-MAX OBSTRUCT-MAX) NIL)).

### 8.886.2 File

Defined in file src/db/friendly.lisp.

## 8.887 Tootsville::Make-Wind-Vector

### 8.887.1 Function

Make-Wind-Vector names an undocumented function, with lambda list (&KEY ((X-MAGNITUDE X-MAGNITUDE) NIL) ((Y-MAGNITUDE Y-MAGNITUDE) NIL)).

### 8.887.2 File

Defined in file src/weather/weather.lisp.

## **8.888 Tootsville::Make-Wind-Vector-Field**

### **8.888.1 Function**

Make-Wind-Vector-Field names a function, with lambda list NIL:

Create the wind vector field for the entire island of Tootanga.

### **8.888.2 File**

Defined in file src/weather/weather.lisp.

## 8.889 Tootsville::Make-World

### 8.889.1 Function

Make-World names an undocumented function, with lambda list (&KEY ((MONIKER MONIKER) NIL) ((NAME NAME) NIL)).

### 8.889.2 File

Defined in file src/db/friendly.lisp.

## 8.890 Tootsville::Make-Wtl-Course

### 8.890.1 Function

Make-Wtl-Course names an undocumented function, with lambda list (&KEY ((SPEED SPEED) NIL) ((START-TIME START-TIME) NIL) ((END-TIME END-TIME) NIL) ((START-POINT START-POINT) NIL) ((END-POINT END-POINT) NIL) ((LATITUDE LATITUDE) NIL) ((LONGITUDE LONGITUDE) NIL) ((ALTITUDE ALTITUDE) NIL) ((WORLD WORLD) NIL) ((FACING FACING) NIL)).

### 8.890.2 File

Defined in file src/characters/robots.lisp.

## 8.891 Tootsville::Map-Places

### 8.891.1 Type

Map-Places names a TYPE:

A symbol representing one of the planes in which the game takes place.

CHOR

Choerogryllum

MOON

The Moon

OTHM

The Other Moon

PINK

The Pink Moon

ORBIT

In space, in transit between Choerogryllum and The Moon.

## 8.892 Tootsville::Maybe-Parent-Approval

### 8.892.1 Function

Maybe-Parent-Approval names a function, with lambda list (TOOT CLIENT):

Check for existing parent approval.

If a parent has already authorized this Toot, they'll sign right in.

Calls Section 8.1388 [TOOTSVILLE WS-APPROVE-TOOT], page 1675, or Section 8.1393 [TOOTSVILLE WS-DENY-TOOT], page 1680, if an existing approval exists. Otherwise, returns silently.

### 8.892.2 File

Defined in file src/websockets.lisp.

## **8.893 Tootsville::Mayor-Louis-Personality**

### **8.893.1 Class**

Mayor-Louis-Personality names a class, with one superclass: Section 8.1090 [TOOTSVILLE ROBOT-MAYOR-LOUIS], page 1375.

This class defines a character named Mayor-Louis

### **8.893.2 Slots**

Class Mayor-Louis-Personality has no direct slots defined.



## **8.894 Tootsville::Memcached-Get-Key**

### **8.894.1 Function**

Memcached-Get-Key names an undocumented function, with lambda list (KEY).

### **8.894.2 File**

Defined in file src/db/memcached.lisp.

## **8.895 Tootsville::Metronome-Idle-Tasks**

### **8.895.1 Function**

Metronome-Idle-Tasks names a function, with lambda list NIL:

Returns only those Metronome tasks without a live thread.

Also reaps (by joining) finished threads.

See Section 8.1114 [TOOTSVILLE RUN-METRONOME-TASKS], page 1399,

### **8.895.2 File**

Defined in file src/metronome.lisp.

## 8.896 Tootsville::Metronome-Register

### 8.896.1 Function

Metronome-Register names a function, with lambda list (TASK):

Safely register TASK with the metronome.

Most users will prefer Section 8.379 [TOOTSVILLE DO-METRONOME], page 635, for that purpose. See also Section 8.1114 [TOOTSVILLE RUN-METRONOME-TASKS], page 1399, for a discussion of the metronome, and Section 8.897 [TOOTSVILLE METRONOME-REMOVE], page 1182, for the complementary function.

### 8.896.2 File

Defined in file src/metronome.lisp.

## 8.897 Tootsville::Metronome-Remove

### 8.897.1 Function

Metronome-Remove names a function, with lambda list (TASK):

Safely remove TASK from the metronome's schedule.

See Section 8.1114 [TOOTSVILLE RUN-METRONOME-TASKS], page 1399, for a discussion of the metronome; see Section 8.379 [TOOTSVILLE DO-METRONOME], page 635, and Section 8.896 [TOOTSVILLE METRONOME-REGISTER], page 1181, to schedule a task.

### 8.897.2 File

Defined in file src/metronome.lisp.

## 8.898 Tootsville::Metronome-Task

### 8.898.1 Class

Metronome-Task names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.898.2 Slots

Class Metronome-Task has 5 direct slot definitions:

Frequency

One-Shot-Time

Name

Function

Thread

## **8.899 Tootsville::Metronome-Task-Frequency**

### **8.899.1 Function**

Metronome-Task-Frequency names an undocumented function, with lambda list (INSTANCE).

### **8.899.2 File**

Defined in file src/metronome.lisp.

### **8.899.3 SetF Function**

(SETF Metronome-Task-Frequency) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.899.4 File**

Defined in file src/metronome.lisp.

## 8.900 Tootsville::Metronome-Task-Function

### 8.900.1 Function

Metronome-Task-Function names an undocumented function, with lambda list (INSTANCE).

### 8.900.2 File

Defined in file src/metronome.lisp.

### 8.900.3 SetF Function

(SETF Metronome-Task-Function) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.900.4 File

Defined in file src/metronome.lisp.

## **8.901 Tootsville::Metronome-Task-Name**

### **8.901.1 Function**

Metronome-Task-Name names an undocumented function, with lambda list (INSTANCE).

### **8.901.2 File**

Defined in file src/metronome.lisp.

### **8.901.3 SetF Function**

(SETF Metronome-Task-Name) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.901.4 File**

Defined in file src/metronome.lisp.



## 8.902 Tootsville::Metronome-Task-One-Shot-Time

### 8.902.1 Function

Metronome-Task-One-Shot-Time names an undocumented function, with lambda list (INSTANCE).

### 8.902.2 File

Defined in file src/metronome.lisp.

### 8.902.3 SetF Function

(SETF Metronome-Task-One-Shot-Time) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.902.4 File

Defined in file src/metronome.lisp.

## **8.903 Tootsville::Metronome-Task-P**

### **8.903.1 Function**

Metronome-Task-P names an undocumented function, with lambda list (OBJECT).

### **8.903.2 File**

Defined in file src/metronome.lisp.

## **8.904 Tootsville::Metronome-Task-Thread**

### **8.904.1 Function**

Metronome-Task-Thread names an undocumented function, with lambda list (INSTANCE).

### **8.904.2 File**

Defined in file src/metronome.lisp.

### **8.904.3 SetF Function**

(SETF Metronome-Task-Thread) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.904.4 File**

Defined in file src/metronome.lisp.

## 8.905 Tootsville::Mist

### 8.905.1 Class

Mist names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.905.2 Slots

Class Mist has 8 direct slot definitions:

World

Latitude-1

Longitude-1

Altitude-1

Latitude-2

Longitude-2

Altitude-2

Definedp

## **8.906 Tootsville::Mist-Altitude-1**

### **8.906.1 Function**

Mist-Altitude-1 names an undocumented function, with lambda list (INSTANCE).

### **8.906.2 File**

Defined in file src/db/friendly.lisp.

### **8.906.3 SetF Function**

(SETF Mist-Altitude-1) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.906.4 File**

Defined in file src/db/friendly.lisp.

## **8.907 Tootsville::Mist-Altitude-2**

### **8.907.1 Function**

Mist-Altitude-2 names an undocumented function, with lambda list (INSTANCE).

### **8.907.2 File**

Defined in file src/db/friendly.lisp.

### **8.907.3 SetF Function**

(SETF Mist-Altitude-2) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.907.4 File**

Defined in file src/db/friendly.lisp.

## 8.908 Tootsville::Mist-Definedp

### 8.908.1 Function

Mist-Definedp names an undocumented function, with lambda list (INSTANCE).

### 8.908.2 File

Defined in file src/db/friendly.lisp.

### 8.908.3 SetF Function

(SETF Mist-Definedp) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.908.4 File

Defined in file src/db/friendly.lisp.

## **8.909 Tootsville::Mist-Latitude-1**

### **8.909.1 Function**

Mist-Latitude-1 names an undocumented function, with lambda list (INSTANCE).

### **8.909.2 File**

Defined in file src/db/friendly.lisp.

### **8.909.3 SetF Function**

(SETF Mist-Latitude-1) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.909.4 File**

Defined in file src/db/friendly.lisp.



## 8.910 Tootsville::Mist-Latitude-2

### 8.910.1 Function

Mist-Latitude-2 names an undocumented function, with lambda list (INSTANCE).

### 8.910.2 File

Defined in file src/db/friendly.lisp.

### 8.910.3 SetF Function

(SETF Mist-Latitude-2) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.910.4 File

Defined in file src/db/friendly.lisp.

## **8.911 Tootsville::Mist-Longitude-1**

### **8.911.1 Function**

Mist-Longitude-1 names an undocumented function, with lambda list (INSTANCE).

### **8.911.2 File**

Defined in file src/db/friendly.lisp.

### **8.911.3 SetF Function**

(SETF Mist-Longitude-1) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.911.4 File**

Defined in file src/db/friendly.lisp.

## **8.912 Tootsville::Mist-Longitude-2**

### **8.912.1 Function**

Mist-Longitude-2 names an undocumented function, with lambda list (INSTANCE).

### **8.912.2 File**

Defined in file src/db/friendly.lisp.

### **8.912.3 SetF Function**

(SETF Mist-Longitude-2) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.912.4 File**

Defined in file src/db/friendly.lisp.

## **8.913 Tootsville::Mist-P**

### **8.913.1 Function**

Mist-P names an undocumented function, with lambda list (OBJECT).

### **8.913.2 File**

Defined in file src/db/friendly.lisp.

## 8.914 Tootsville::Mist-World

### 8.914.1 Function

Mist-World names an undocumented function, with lambda list (INSTANCE).

### 8.914.2 File

Defined in file src/db/friendly.lisp.

### 8.914.3 SetF Function

(SETF Mist-World) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.914.4 File

Defined in file src/db/friendly.lisp.

## **8.915 Tootsville::Moo-Personality**

### **8.915.1 Class**

Moo-Personality names a class, with one superclass: Section 8.1092 [TOOTSVILLE ROBOT-MOO], page 1377.

This class defines a character named Moo

### **8.915.2 Slots**

Class Moo-Personality has no direct slots defined.

## 8.916 Tootsville::Moon-Position

### 8.916.1 Function

Moon-Position names a function, with lambda list (MOON-OR-PERIOD &OPTIONAL (TIME (GET-UNIVERSAL-TIME))):

Returns the relative position of MOON-OR-PERIOD in the sky at TIME.

Returns the coordinates in  $(x,y,\phi)$  triplet list form, where  $\phi$  represents the phase of the moon.

### 8.916.2 File

Defined in file src/weather/sun-moon.lisp.

## 8.917 Tootsville::Music

### 8.917.1 Class

Music names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.917.2 Slots

Class Music has 6 direct slot definitions:

Id

Title

Artist

Genre

License

Moniker



## 8.918 Tootsville::Music-Artist

### 8.918.1 Function

Music-Artist names an undocumented function, with lambda list (INSTANCE).

### 8.918.2 File

Defined in file src/db/friendly.lisp.

### 8.918.3 SetF Function

(SETF Music-Artist) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.918.4 File

Defined in file src/db/friendly.lisp.

## **8.919 Tootsville::Music-Genre**

### **8.919.1 Function**

Music-Genre names an undocumented function, with lambda list (INSTANCE).

### **8.919.2 File**

Defined in file src/db/friendly.lisp.

### **8.919.3 SetF Function**

(SETF Music-Genre) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.919.4 File**

Defined in file src/db/friendly.lisp.

## **8.920 Tootsville::Music-Id**

### **8.920.1 Function**

Music-Id names an undocumented function, with lambda list (INSTANCE).

### **8.920.2 File**

Defined in file src/db/friendly.lisp.

### **8.920.3 SetF Function**

(SETF Music-Id) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.920.4 File**

Defined in file src/db/friendly.lisp.

## **8.921 Tootsville::Music-License**

### **8.921.1 Function**

Music-License names an undocumented function, with lambda list (INSTANCE).

### **8.921.2 File**

Defined in file src/db/friendly.lisp.

### **8.921.3 SetF Function**

(SETF Music-License) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.921.4 File**

Defined in file src/db/friendly.lisp.

## 8.922 Tootsville::Music-Moniker

### 8.922.1 Function

Music-Moniker names an undocumented function, with lambda list (INSTANCE).

### 8.922.2 File

Defined in file src/db/friendly.lisp.

### 8.922.3 SetF Function

(SETF Music-Moniker) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.922.4 File

Defined in file src/db/friendly.lisp.

## **8.923 Tootsville::Music-P**

### **8.923.1 Function**

Music-P names an undocumented function, with lambda list (OBJECT).

### **8.923.2 File**

Defined in file src/db/friendly.lisp.

## 8.924 Tootsville::Music-Title

### 8.924.1 Function

Music-Title names an undocumented function, with lambda list (INSTANCE).

### 8.924.2 File

Defined in file src/db/friendly.lisp.

### 8.924.3 SetF Function

(SETF Music-Title) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.924.4 File

Defined in file src/db/friendly.lisp.

## 8.925 Tootsville::Name-For-Content-Type

### 8.925.1 Function

Name-For-Content-Type names a function, with lambda list (CONTENT-TYPE):

Get the name to be used in function names for CONTENT-TYPE.

Typically this is the file extension, but if none is known, it's the end of the CONTENT-TYPE after the slash.

### 8.925.2 File

Defined in file src/web.lisp.



## 8.926 Tootsville::Name-Idle-Threads-Sequentially

### 8.926.1 Function

Name-Idle-Threads-Sequentially names a function, with lambda list (COUNT):

Name all of the idle asynchronous worker threads with numbers up to COUNT.

### 8.926.2 File

Defined in file src/main.lisp.

## **8.927 Tootsville::Nearp**

### **8.927.1 Function**

Nearp names a function, with lambda list (THING PLACE):

Is THING near to PLACE?

“Near,” in this case, means “close enough to observe actions at PLACE.” Network events are not propagated to observers who are not NEARP to the event being observed.

### **8.927.2 File**

Defined in file src/characters/robots.lisp.

## **8.928 Tootsville::Nevermind-Personality**

### **8.928.1 Class**

Nevermind-Personality names a class, with one superclass: Section 8.1093 [TOOTSVILLE ROBOT-NEVERMIND], page 1378.

This class defines a character named Nevermind

### **8.928.2 Slots**

Class Nevermind-Personality has no direct slots defined.

## **8.929 Tootsville::Normalize-Url**

### **8.929.1 Function**

Normalize-Url names a function, with lambda list (URL):

Normalize URL into a canonical form, using some typical UNIX pathname rules.

### **8.929.2 File**

Defined in file `src/endpoints/gossip/alex/alex.lisp`.

## 8.930 Tootsville::Not-Found

### 8.930.1 Class

Not-Found names a class, with one superclass: Section 8.667 [TOOTSVILLE HTTP-CLIENT-ERROR], page 925.

Some object could not be found based on the identification provided.

### 8.930.2 Slots

Class Not-Found has 2 direct slot definitions:

`Http-Status-Code`

`Thing`

## **8.931 Tootsville::Not-Found-If-Null**

### **8.931.1 Function**

Not-Found-If-Null names a function, with lambda list (THING):

If THING is null, then abort with a 404 Not Found.

### **8.931.2 File**

Defined in file src/acceptor.lisp.

## **8.932 Tootsville::Not-Found-Thing**

### **8.932.1 Function**

Not-Found-Thing names an undocumented function, with lambda list (CONDITION).

### **8.932.2 SetF Function**

(SETF Not-Found-Thing) names an undocumented function, with lambda list (NEW-VALUE CONDITION).

## 8.933 Tootsville::Not-Your-Toot-Error

### 8.933.1 Class

Not-Your-Toot-Error names a class, with one superclass: Section 8.667 [TOOTSVILLE HTTP-CLIENT-ERROR], page 925.

An error thrown when a player tries to alter another player's Toot

### 8.933.2 Slots

Class Not-Your-Toot-Error has 2 direct slot definitions:

Http-Status-Code  
Name



## **8.934 Tootsville::Null-If-Empty**

### **8.934.1 Function**

Null-If-Empty names an undocumented function, with lambda list (STRING).

### **8.934.2 File**

Defined in file src/version.lisp.

## **8.935 Tootsville::On-Exception**

### **8.935.1 Function**

On-Exception names an undocumented function, with lambda list (CODE).

## **8.936 Tootsville::Open-Log-File**

### **8.936.1 Function**

Open-Log-File names a function, with lambda list (PATHNAME):

Open PATHNAME for logging.

### **8.936.2 File**

Defined in file src/logging.lisp.

## **8.937 Tootsville::Pad-To-Multiple-Of-8**

### **8.937.1 Function**

Pad-To-Multiple-Of-8 names an undocumented function, with lambda list (STRING).

### **8.937.2 File**

Defined in file `src/auth/auth-firebase.lisp`.

## 8.938 Tootsville::Parent-Child

### 8.938.1 Class

Parent-Child names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.938.2 Slots

Class Parent-Child has 2 direct slot definitions:

Parent

Child

## **8.939 Tootsville::Parent-Child-Child**

### **8.939.1 Function**

Parent-Child-Child names an undocumented function, with lambda list (INSTANCE).

### **8.939.2 File**

Defined in file src/db/friendly.lisp.

### **8.939.3 SetF Function**

(SETF Parent-Child-Child) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.939.4 File**

Defined in file src/db/friendly.lisp.

## **8.940 Tootsville::Parent-Child-P**

### **8.940.1 Function**

Parent-Child-P names an undocumented function, with lambda list (OBJECT).

### **8.940.2 File**

Defined in file src/db/friendly.lisp.

## **8.941 Tootsville::Parent-Child-Parent**

### **8.941.1 Function**

Parent-Child-Parent names an undocumented function, with lambda list (INSTANCE).

### **8.941.2 File**

Defined in file src/db/friendly.lisp.

### **8.941.3 SetF Function**

(SETF Parent-Child-Parent) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.941.4 File**

Defined in file src/db/friendly.lisp.



## 8.942 Tootsville::Parent-Deny-Permission

### 8.942.1 Function

Parent-Deny-Permission names a function, with lambda list (REQUEST &KEY (VIA web)):

The parent who was given REQUEST has denied permission via VIA.

The child who placed REQUEST is *not* being given permission to play in Tootsville.

Returns NIL

### 8.942.2 File

Defined in file src/users.lisp.

## 8.943 Tootsville::Parent-Grant-Permission

### 8.943.1 Function

Parent-Grant-Permission names a function, with lambda list (REQUEST &KEY (HOURS 168) (VIA web)):

The parent \*USER\* grants REQUEST for HOURS via VIA.

This sets the approval time to LOCAL-TIME::NOW (not in this manual) and allows HOURS of play time from LOCAL-TIME::NOW (not in this manual). The Section 8.207 [TOOTSVILLE CHILD-REQUEST-RESPONSE], page 461, of REQUEST is set to an explanation that \*USER\* approved the request via VIA. VIA can contain further comments, which will be presented in the UI.

Returns NIL.

### 8.943.2 File

Defined in file src/users.lisp.

## 8.944 Tootsville::Parse-Backtrace

### 8.944.1 Function

Parse-Backtrace names a function, with lambda list (BT):

Break lines of a backtrace into error messag, date/time, and call frames (stack)

### 8.944.2 File

Defined in file src/errors.lisp.

## 8.945 Tootsville::Parse-Color24

### 8.945.1 Function

Parse-Color24 names a function, with lambda list (COLOR):

Return a Section 8.223 [TOOTSVILLE COLOR24], page 477, object for the color designator COLOR.

Parse COLOR as a name for a color, or a hex 24-bit color value. It can also be an HTML-style or CSS-style value.

Syntax can be:

- A Section 8.223 [TOOTSVILLE COLOR24], page 477, object, which is returned unchanged.
- An integer of 24 bits, in which the upper 8 bits represent the red channel, the middle 8 bits the green, and the lower 8 bits the blue channel.
- A color name from the list Section 8.84 [TOOTSVILLE +COLOR24-NAMES+], page 338. Spaces are interchangeable with hyphens and the value is not case-sensitive.
- An RGB byte-value triplet as in CSS, of the form `rgb(R,G,B)`, where R, G, and B are unsigned 8-bit decimal integers, i.e. values from 0-255.
- An HTML-style hex color code in the form `#RGB` or `#RRGGBB`. When only three hex digits are provided, each is doubled to form an unsigned 8-bit value; thus, `#abc` is the same as `#aabbcc`.
- A set of 6 hex digits of the same form `rrggbb` as the HTML form `#rrggbb`, without any sigil.

### 8.945.2 File

Defined in file `src/types/color+pattern.lisp`.

## 8.946 Tootsville::Parse-Operator-Command

### 8.946.1 Function

Parse-Operator-Command names a function, with lambda list (STRING):

Parse an operator command in STRING (beginning with #)

### 8.946.2 File

Defined in file src/infinity/legacy-commands.lisp.

## 8.947 Tootsville::Parse-Uri-As-Template

### 8.947.1 Function

Parse-Uri-As-Template names a function, with lambda list (URI):

Parse URI into a template list.

URI is a series of path elements joined by ‘/’ characters. Each path element can be a constant string, or a variable. Variable terms begin with ‘:’ characters; string constant terms do not.

Returns a list in which variable terms are keywords and constant terms are strings.

### 8.947.2 File

Defined in file src/endpoint.lisp.

## 8.948 Tootsville::Parse-Wtl-For-Robot

### 8.948.1 Function

Parse-Wtl-For-Robot names a function, with lambda list (WTL):

Parse the WTL JSON into a WTL-Course structure

### 8.948.2 File

Defined in file src/characters/robots.lisp.

## 8.949 Tootsville::Path->Openapi

### 8.949.1 Function

Path->Openapi names a function, with lambda list (ENDPOINT-GROUP):

Given a path list ENDPOINT-GROUP, return an OpenAPI URI string.

The path list ENDPOINT-GROUP consists a URI template of constant strings and variables as symbols and a list of endpoints which share that template, each of which is a PList with a :METHOD, :TEMPLATE, :CONTENT-TYPE, :FN, and :DOCSTRING.

### 8.949.2 File

Defined in file src/endpoints/slash-meta-game.lisp.



## 8.950 Tootsville::Pattern

### 8.950.1 Class

Pattern names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.950.2 Slots

Class Pattern has 2 direct slot definitions:

Id

Name

## **8.951 Tootsville::Pattern-Id**

### **8.951.1 Function**

Pattern-Id names an undocumented function, with lambda list (INSTANCE).

### **8.951.2 File**

Defined in file src/db/friendly.lisp.

### **8.951.3 SetF Function**

(SETF Pattern-Id) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.951.4 File**

Defined in file src/db/friendly.lisp.

## 8.952 Tootsville::Pattern-Name

### 8.952.1 Function

Pattern-Name names an undocumented function, with lambda list (INSTANCE).

### 8.952.2 File

Defined in file src/db/friendly.lisp.

### 8.952.3 SetF Function

(SETF Pattern-Name) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.952.4 File

Defined in file src/db/friendly.lisp.

## **8.953 Tootsville::Pattern-P**

### **8.953.1 Function**

Pattern-P names an undocumented function, with lambda list (OBJECT).

### **8.953.2 File**

Defined in file src/db/friendly.lisp.

## **8.954 Tootsville::Peer-Address**

### **8.954.1 Function**

Peer-Address names an undocumented function, with lambda list (TOOT).

## **8.955 Tootsville::Pending-Child-Approval-Request**

### **8.955.1 Function**

Pending-Child-Approval-Request names an undocumented function, with lambda list (USER).

### **8.955.2 File**

Defined in file src/users.lisp.

## **8.956 Tootsville::Pending-Child-Requests-By-Toot**

### **8.956.1 Function**

Pending-Child-Requests-By-Toot names an undocumented function, with lambda list (TOOT).

### **8.956.2 File**

Defined in file src/users.lisp.

## 8.957 Tootsville::Person

### 8.957.1 Class

Person names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.957.2 Slots

Class Person has 9 direct slot definitions:

Uuid

Display-Name

Given-Name

Surname

Date-Of-Birth

Age

Sensitivep

Gender

Lang



## 8.958 Tootsville::Person-Age

### 8.958.1 Function

Person-Age names an undocumented function, with lambda list (INSTANCE).

### 8.958.2 File

Defined in file src/db/friendly.lisp.

### 8.958.3 SetF Function

(SETF Person-Age) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.958.4 File

Defined in file src/db/friendly.lisp.

## **8.959 Tootsville::Person-Age\***

### **8.959.1 Function**

Person-Age\* names a function, with lambda list (&OPTIONAL (USER \*USER\*)):

Get a person's age in years.

### **8.959.2 File**

Defined in file src/users.lisp.

## **8.960 Tootsville::Person-Date-Of-Birth**

### **8.960.1 Function**

Person-Date-Of-Birth names an undocumented function, with lambda list (INSTANCE).

### **8.960.2 File**

Defined in file src/db/friendly.lisp.

### **8.960.3 SetF Function**

(SETF Person-Date-Of-Birth) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.960.4 File**

Defined in file src/db/friendly.lisp.

## **8.961 Tootsville::Person-Display-Name**

### **8.961.1 Function**

Person-Display-Name names an undocumented function, with lambda list (INSTANCE).

### **8.961.2 File**

Defined in file src/db/friendly.lisp.

### **8.961.3 SetF Function**

(SETF Person-Display-Name) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.961.4 File**

Defined in file src/db/friendly.lisp.

## 8.962 Tootsville::Person-First-Email

### 8.962.1 Function

Person-First-Email names a function, with lambda list (&OPTIONAL (USER \*USER\*)):

Gives one possible eMail address associated with USER.

Uses the first, alphabetically speaking.

### 8.962.2 File

Defined in file src/users.lisp.

## **8.963 Tootsville::Person-Gender**

### **8.963.1 Function**

Person-Gender names an undocumented function, with lambda list (INSTANCE).

### **8.963.2 File**

Defined in file src/db/friendly.lisp.

### **8.963.3 SetF Function**

(SETF Person-Gender) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.963.4 File**

Defined in file src/db/friendly.lisp.

## **8.964 Tootsville::Person-Given-Name**

### **8.964.1 Function**

Person-Given-Name names an undocumented function, with lambda list (INSTANCE).

### **8.964.2 File**

Defined in file src/db/friendly.lisp.

### **8.964.3 SetF Function**

(SETF Person-Given-Name) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.964.4 File**

Defined in file src/db/friendly.lisp.

## 8.965 Tootsville::Person-Info

### 8.965.1 Function

Person-Info names a function, with lambda list (&OPTIONAL (USER \*USER\*)):

Creates a JSON-like PList of information about USER.

Its contents are:

<code>uuid</code>	The person's UUID
<code>displayName</code>	The person's name, formatted for display.
<code>patronP</code>	True if this person is a patron of the CIWTA project.
<code>gender</code>	One of (unknown/other), (female), or (male).
<code>givenName</code>	The person's given name.
<code>surname</code>	The person's surname.
<code>language</code>	The person's spoken language
<code>sensitiveP</code>	If true, this is a Sensitive Player
<code>dateOfBirth</code>	The person's date of birth, in an ISO format string
<code>age</code>	The person's age in years

### 8.965.2 File

Defined in file `src/users.lisp`.



## **8.966 Tootsville::Person-Is-Patron-P**

### **8.966.1 Function**

Person-Is-Patron-P names a function, with lambda list (PERSON):

Returns true if PERSON is a patron of CIWTA.

Currently just me.

### **8.966.2 File**

Defined in file src/users.lisp.

## **8.967 Tootsville::Person-Lang**

### **8.967.1 Function**

Person-Lang names an undocumented function, with lambda list (INSTANCE).

### **8.967.2 File**

Defined in file src/db/friendly.lisp.

### **8.967.3 SetF Function**

(SETF Person-Lang) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.967.4 File**

Defined in file src/db/friendly.lisp.

## 8.968 Tootsville::Person-Link

### 8.968.1 Class

Person-Link names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.968.2 Slots

Class Person-Link has 6 direct slot definitions:

Uuid

Person

Rel

Url

Label

Provenance

## **8.969 Tootsville::Person-Link-Label**

### **8.969.1 Function**

Person-Link-Label names an undocumented function, with lambda list (INSTANCE).

### **8.969.2 File**

Defined in file src/db/friendly.lisp.

### **8.969.3 SetF Function**

(SETF Person-Link-Label) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.969.4 File**

Defined in file src/db/friendly.lisp.

## **8.970 Tootsville::Person-Link-P**

### **8.970.1 Function**

Person-Link-P names an undocumented function, with lambda list (OBJECT).

### **8.970.2 File**

Defined in file src/db/friendly.lisp.

## **8.971 Tootsville::Person-Link-Person**

### **8.971.1 Function**

Person-Link-Person names an undocumented function, with lambda list (INSTANCE).

### **8.971.2 File**

Defined in file src/db/friendly.lisp.

### **8.971.3 SetF Function**

(SETF Person-Link-Person) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.971.4 File**

Defined in file src/db/friendly.lisp.

## 8.972 Tootsville::Person-Link-Provenance

### 8.972.1 Function

Person-Link-Provenance names an undocumented function, with lambda list (INSTANCE).

### 8.972.2 File

Defined in file src/db/friendly.lisp.

### 8.972.3 SetF Function

(SETF Person-Link-Provenance) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.972.4 File

Defined in file src/db/friendly.lisp.

## **8.973 Tootsville::Person-Link-Rel**

### **8.973.1 Function**

Person-Link-Rel names an undocumented function, with lambda list (INSTANCE).

### **8.973.2 File**

Defined in file src/db/friendly.lisp.

### **8.973.3 SetF Function**

(SETF Person-Link-Rel) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.973.4 File**

Defined in file src/db/friendly.lisp.



## 8.974 Tootsville::Person-Link-Url

### 8.974.1 Function

Person-Link-Url names an undocumented function, with lambda list (INSTANCE).

### 8.974.2 File

Defined in file src/db/friendly.lisp.

### 8.974.3 SetF Function

(SETF Person-Link-Url) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.974.4 File

Defined in file src/db/friendly.lisp.

## **8.975 Tootsville::Person-Link-Uuid**

### **8.975.1 Function**

Person-Link-Uuid names an undocumented function, with lambda list (INSTANCE).

### **8.975.2 File**

Defined in file src/db/friendly.lisp.

### **8.975.3 SetF Function**

(SETF Person-Link-Uuid) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.975.4 File**

Defined in file src/db/friendly.lisp.

## **8.976 Tootsville::Person-Links-To-Email**

### **8.976.1 Function**

Person-Links-To-Email names an undocumented function, with lambda list (EMAIL).

### **8.976.2 File**

Defined in file src/users.lisp.

## **8.977 Tootsville::Person-P**

### **8.977.1 Function**

Person-P names an undocumented function, with lambda list (OBJECT).

### **8.977.2 File**

Defined in file src/db/friendly.lisp.

## **8.978 Tootsville::Person-Sensitivep**

### **8.978.1 Function**

Person-Sensitivep names an undocumented function, with lambda list (INSTANCE).

### **8.978.2 File**

Defined in file src/db/friendly.lisp.

### **8.978.3 SetF Function**

(SETF Person-Sensitivep) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.978.4 File**

Defined in file src/db/friendly.lisp.

## **8.979 Tootsville::Person-Surname**

### **8.979.1 Function**

Person-Surname names an undocumented function, with lambda list (INSTANCE).

### **8.979.2 File**

Defined in file src/db/friendly.lisp.

### **8.979.3 SetF Function**

(SETF Person-Surname) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.979.4 File**

Defined in file src/db/friendly.lisp.

## 8.980 Tootsville::Person-Uuid

### 8.980.1 Function

Person-Uuid names an undocumented function, with lambda list (INSTANCE).

### 8.980.2 File

Defined in file src/db/friendly.lisp.

### 8.980.3 SetF Function

(SETF Person-Uuid) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.980.4 File

Defined in file src/db/friendly.lisp.

## **8.981 Tootsville::Picasso-Personality**

### **8.981.1 Class**

Picasso-Personality names a class, with one superclass: Section 8.1094 [TOOTSVILLE ROBOT-PICASSO], page 1379.

This class defines a character named Picasso

### **8.981.2 Slots**

Class Picasso-Personality has no direct slots defined.



## 8.982 Tootsville::Place

### 8.982.1 Class

Place names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.982.2 Slots

Class Place has 9 direct slot definitions:

Uuid

World

Latitude

Longitude

Altitude

Shape

Kind

Attributes

Appearance

## **8.983 Tootsville::Place-Altitude**

### **8.983.1 Function**

Place-Altitude names an undocumented function, with lambda list (INSTANCE).

### **8.983.2 File**

Defined in file src/db/friendly.lisp.

### **8.983.3 SetF Function**

(SETF Place-Altitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.983.4 File**

Defined in file src/db/friendly.lisp.

## 8.984 Tootsville::Place-Appearance

### 8.984.1 Function

Place-Appearance names an undocumented function, with lambda list (INSTANCE).

### 8.984.2 File

Defined in file src/db/friendly.lisp.

### 8.984.3 SetF Function

(SETF Place-Appearance) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.984.4 File

Defined in file src/db/friendly.lisp.

## **8.985 Tootsville::Place-Attributes**

### **8.985.1 Function**

Place-Attributes names an undocumented function, with lambda list (INSTANCE).

### **8.985.2 File**

Defined in file src/db/friendly.lisp.

### **8.985.3 SetF Function**

(SETF Place-Attributes) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.985.4 File**

Defined in file src/db/friendly.lisp.

## **8.986 Tootsville::Place-Furniture**

### **8.986.1 Function**

Place-Furniture names an undocumented function, with lambda list (SLOT X Y Z FACING).

### **8.986.2 File**

Defined in file `src/infinity/legacy-commands.lisp`.

## **8.987 Tootsville::Place-Kind**

### **8.987.1 Function**

Place-Kind names an undocumented function, with lambda list (INSTANCE).

### **8.987.2 File**

Defined in file src/db/friendly.lisp.

### **8.987.3 SetF Function**

(SETF Place-Kind) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.987.4 File**

Defined in file src/db/friendly.lisp.

## 8.988 Tootsville::Place-Latitude

### 8.988.1 Function

Place-Latitude names an undocumented function, with lambda list (INSTANCE).

### 8.988.2 File

Defined in file src/db/friendly.lisp.

### 8.988.3 SetF Function

(SETF Place-Latitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.988.4 File

Defined in file src/db/friendly.lisp.

## **8.989 Tootsville::Place-Longitude**

### **8.989.1 Function**

Place-Longitude names an undocumented function, with lambda list (INSTANCE).

### **8.989.2 File**

Defined in file src/db/friendly.lisp.

### **8.989.3 SetF Function**

(SETF Place-Longitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.989.4 File**

Defined in file src/db/friendly.lisp.



## **8.990 Tootsville::Place-P**

### **8.990.1 Function**

Place-P names an undocumented function, with lambda list (OBJECT).

### **8.990.2 File**

Defined in file src/db/friendly.lisp.

## **8.991 Tootsville::Place-Shape**

### **8.991.1 Function**

Place-Shape names an undocumented function, with lambda list (INSTANCE).

### **8.991.2 File**

Defined in file src/db/friendly.lisp.

### **8.991.3 SetF Function**

(SETF Place-Shape) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.991.4 File**

Defined in file src/db/friendly.lisp.

## 8.992 Tootsville::Place-String

### 8.992.1 Function

Place-String names a function, with lambda list (PLACE):

Formats PLACE in the encoding for the client.

The PLACE is encoded into a string in the form:

```
kind:shape|appearance|attributes
```

### 8.992.2 File

Defined in file src/items.lisp.

## 8.993 Tootsville::Place-String-Circle

### 8.993.1 Function

Place-String-Circle names a function, with lambda list (RADIUS X-CENTER Z-CENTER SEGMENTS):

Defines a place-string for a circle of RADIUS centered at X-CENTER, Z-CENTER with SEGMENTS precision.

An n-sided (SEGMENTS-sided) regular polygon approximating a circle will be created at (X-CENTER, Z-CENTER) and returned as the path segments string used by the client; i.e. a list of the form  $x,y,z\tilde{x},y,z\tilde{x},y,z$  with  $\tilde{}$  delimiters between coordinate lists joined by  $,$ .

### 8.993.2 File

Defined in file src/items.lisp.

## **8.994 Tootsville::Place-Uuid**

### **8.994.1 Function**

Place-Uuid names an undocumented function, with lambda list (INSTANCE).

### **8.994.2 File**

Defined in file src/db/friendly.lisp.

### **8.994.3 SetF Function**

(SETF Place-Uuid) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.994.4 File**

Defined in file src/db/friendly.lisp.

## **8.995 Tootsville::Place-World**

### **8.995.1 Function**

Place-World names an undocumented function, with lambda list (INSTANCE).

### **8.995.2 File**

Defined in file src/db/friendly.lisp.

### **8.995.3 SetF Function**

(SETF Place-World) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.995.4 File**

Defined in file src/db/friendly.lisp.

## 8.996 Tootsville::Places-At-Position

### 8.996.1 Function

Places-At-Position names a function, with lambda list (WORLD LAT LONG ALT):

Returns all Places at WORLD at LAT-itude, LONG-itude, ALT-itude.

See Section 8.702 [TOOTSVILLE INFINITY-GET-ROOM-VARS], page 969, for a discussion of the Place system.

### 8.996.2 File

Defined in file src/infinity/new-commands-20.lisp.

## 8.997 Tootsville::Play-With-Toot

### 8.997.1 Function

Play-With-Toot names a function, with lambda list (TOOT):

Set up the \*USER\* to play with Toot object TOOT.

Performs announcement of the player to the world and other bookkeeping.

See Section 8.718 [TOOTSVILLE INFINITY-PLAY-WITH], page 989.

The client will receive a minor broadcast storm of information about their Toot and the game world. This will, at a minimum, include a success message from `playWith`, their own avatar information, nearby players' avatar information, and Section 8.819 [TOOTSVILLE LOCAL-ROOM-VARS], page 1104, for their immediate vicinity.

### 8.997.2 File

Defined in file `src/infinity/new-commands-20.lisp`.



## 8.998 Tootsville::Player-Adultp

### 8.998.1 Function

Player-Adultp names an undocumented function, with lambda list (&OPTIONAL (PLAYER \*USER\*)).

### 8.998.2 File

Defined in file src/users.lisp.

## **8.999 Tootsville::Player-Alert**

### **8.999.1 Function**

Player-Alert names a function, with lambda list (PERSON &REST MESSAGE):

Sends an asynchronous notification alert MESSAGE to PERSON

### **8.999.2 File**

Defined in file src/users.lisp.

## **8.1000 Tootsville::Player-Childp**

### **8.1000.1 Function**

Player-Childp names an undocumented function, with lambda list (&OPTIONAL (PLAYER \*USER\*)).

### **8.1000.2 File**

Defined in file src/users.lisp.

## **8.1001 Tootsville::Player-Toots**

### **8.1001.1 Function**

Player-Toots names an undocumented function, with lambda list (&OPTIONAL (PLAYER \*USER\*)).

### **8.1001.2 File**

Defined in file src/users.lisp.

## **8.1002 Tootsville::Plist-To-English**

### **8.1002.1 Function**

Plist-To-English names an undocumented function, with lambda list (PLIST).

### **8.1002.2 File**

Defined in file `src/endpoints/slash-users.lisp`.

## **8.1003 Tootsville::Plist-With-Index**

### **8.1003.1 Function**

Plist-With-Index names a function, with lambda list (LIST):

Zip LIST with sequential numbers from 0, creating a plist whose keys are sequential integers.

### **8.1003.2 File**

Defined in file `src/infinity/new-commands-20.lisp`.

## 8.1004 Tootsville::Point-Underwater-P

### 8.1004.1 Function

Point-Underwater-P names a function, with lambda list (LATITUDE LONGITUDE):

Is the point underwater? TODO

### 8.1004.2 File

Defined in file src/terrain.lisp.

## **8.1005 Tootsville::Post-Sign-In**

### **8.1005.1 Function**

Post-Sign-In names a function, with lambda list (USER):

Perform housekeeping after an user signs in.

This might include sending a pending child prompt.

### **8.1005.2 File**

Defined in file src/users.lisp.



## **8.1006 Tootsville::Post/ Read-Version-Page**

### **8.1006.1 Function**

Post/ Read-Version-Page names a function, with lambda list (PORT):

Power-On-Self-Test: Checks that the server can respond to the version-page query locally.

### **8.1006.2 File**

Defined in file src/power-on-self-test.lisp.

## 8.1007 Tootsville::Potential-Toot-Name-Character-P

### 8.1007.1 Function

Potential-Toot-Name-Character-P names an undocumented function, with lambda list (&REST ARGUMENTS).

### 8.1007.2 File

Defined in file `quicklisp/dists/quicklisp/software/fare-memoization-20180430-git/memoization.lisp`.

## 8.1008 Tootsville::Potential-Toot-Name-P

### 8.1008.1 Function

Potential-Toot-Name-P names a function, with lambda list (TOOT-NAME):

Could TOOT-NAME be allowed as a Toot name?

Toot names must be:

- From three to 32 characters in length, inclusive.
- Characters must be Section 8.1007 [TOOTSVILLE POTENTIAL-TOOT-NAME-CHARACTER-P], page 1292, ie, alphanumeric, or hyphen.
- The first character must be alphabetic
- There can not be two punctuation marks (or spaces) in a row
- There can not be three of the same character in a row, or two hyphens in a row.
- There can not be more than three digits
- Digits must appear only at the end – i.e., if there are any digits, the leftmost digit must be after the rightmost non-digit character.

### 8.1008.2 File

Defined in file `src/types/toot-names.lisp`.

## **8.1009 Tootsville::Power-On-Self-Test**

### **8.1009.1 Function**

Power-On-Self-Test names a function, with lambda list (&KEY (EXITP NIL)):

Perform some sanity checking as a part of testing.

This testing should be much more complete than it really is — it will need to be expanded a great deal to increase confidence in these tests.

### **8.1009.2 File**

Defined in file src/power-on-self-test.lisp.

## 8.1010 Tootsville::Powerset

### 8.1010.1 Function

Powerset names a function, with lambda list (LIST):

Create a powerset of the unordered elements of LIST.

```
(powerset '(a b c))  
((:A :B :C) (:B :C) (:A :C) (:C) (:A :B) (:B) (:A) NIL)
```

### 8.1010.2 File

Defined in file src/db/memcached.lisp.

## **8.1011 Tootsville::Pre-Login-Commands**

### **8.1011.1 Function**

Pre-Login-Commands names an undocumented function, with lambda list (OBJECT).

### **8.1011.2 SetF Function**

(SETF Pre-Login-Commands) names an undocumented function, with lambda list (NEW-VALUE OBJECT).

## **8.1012 Tootsville::Precipitation**

### **8.1012.1 Function**

Precipitation names a function, with lambda list (X Y Z):

The current precipitation at X,Y,Z

### **8.1012.2 File**

Defined in file src/weather/weather.lisp.

## **8.1013 Tootsville::Pretty-Print-Html-Error**

### **8.1013.1 Function**

Pretty-Print-Html-Error names a function, with lambda list (CONDITION):

Produces an HTML page explaining CONDITION.

TODO: Use templates, filter backtrace like Rollbar, do better.

### **8.1013.2 File**

Defined in file src/types/http-types.lisp.



## 8.1014 Tootsville::Print-Help

### 8.1014.1 Function

Print-Help names a function, with lambda list NIL:

Prints a short usage summary to \*STANDARD-OUTPUT\*. Note that this is invoked by calling the program with “help” as its first argument, explicitly — the default behaviour is to run as a FastCGI server.

### 8.1014.2 File

Defined in file src/command-line.lisp.

## 8.1015 Tootsville::Private-Admin-Message

### 8.1015.1 Function

Private-Admin-Message names a function, with lambda list (TITLE MESSAGE &KEY (LABEL TITLE) (USER (ACTIVE-PLAYER))):

Send a unicast admin MESSAGE to USER with TITLE and LABEL.

Instead logs the contents to the console if USER is not connected.

Note that the current Tootsville V client does not make use of LABEL.

### 8.1015.2 File

Defined in file src/websockets.lisp.

## **8.1016 Tootsville::Prod**

### **8.1016.1 Variable**

Prod names an undocumented variable with the value NIL

## **8.1017 Tootsville::Props-Personality**

### **8.1017.1 Class**

Props-Personality names a class, with one superclass: Section 8.1096 [TOOTSVILLE ROBOT-PROPS], page 1381.

This class defines a character named Props

### **8.1017.2 Slots**

Class Props-Personality has no direct slots defined.

## **8.1018 Tootsville::Pull-Records**

### **8.1018.1 Function**

Pull-Records names an undocumented function, with lambda list (NAME).

### **8.1018.2 File**

Defined in file src/db/db-central.lisp.

## **8.1019 Tootsville::Pull-Records-Cache**

### **8.1019.1 Variable**

Pull-Records-Cache names an undocumented variable with the value NIL

## **8.1020 Tootsville::Qa**

### **8.1020.1 Variable**

Qa names an undocumented variable with the value NIL

## **8.1021 Tootsville::Quaestor-Cancel-Event**

### **8.1021.1 Function**

Quaestor-Cancel-Event names a function, with lambda list (EVENT):

Cancel EVENT.

See Section 8.689 [TOOTSVILLE INFINITY-END-EVENT], page 949, for details of the procedure.

### **8.1021.2 File**

Defined in file src/quaestor.lisp.



## 8.1022 Tootsville::Quaestor-Complete-Event

### 8.1022.1 Function

Quaestor-Complete-Event names a function, with lambda list (EVENT SCORE &OPTIONAL MEDAL):

Complete EVENT with SCORE and MEDAL earned.

See Section 8.689 [TOOTSVILLE INFINITY-END-EVENT], page 949, for details of the procedure.

If EVENT is a purchase, then purchase the associated store item; otherwise, perform whatever specific event side-effects are related to the item template.

### 8.1022.2 File

Defined in file src/quaestor.lisp.

## 8.1023 Tootsville::Quaestor-End-Fountain

### 8.1023.1 Function

Quaestor-End-Fountain names a function, with lambda list (EVENT SCORE):

End a fountain EVENT with the user-supplied SCORE.

The SCORE and the (Choerogyllum) day of the week are used to compute the actual number of peanuts earned. See Section 8.242 [TOOTSVILLE COMPUTE-FOUNTAIN-PEANUTS-FOR-SCORE], page 496.

Occasionally, fairy dust is also awarded. See Section 8.243 [TOOTSVILLE COMPUTE-FOUNTAIN-RANDOM-FAIRY-DUST], page 497.

### 8.1023.2 File

Defined in file src/quaestor.lisp.

## 8.1024 Tootsville::Quaestor-Event

### 8.1024.1 Class

Quaestor-Event names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.1024.2 Slots

Class Quaestor-Event has 11 direct slot definitions:

Uuid

Source

Started-By

Started-At

Ended-At

Completedp

Peanuts

Fairy-Dust

Item

Score

Medal

## **8.1025 Tootsville::Quaestor-Event-Completedp**

### **8.1025.1 Function**

Quaestor-Event-Completedp names an undocumented function, with lambda list (INSTANCE).

### **8.1025.2 File**

Defined in file src/db/friendly.lisp.

### **8.1025.3 SetF Function**

(SETF Quaestor-Event-Completedp) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1025.4 File**

Defined in file src/db/friendly.lisp.

## 8.1026 Tootsville::Quaestor-Event-Ended-At

### 8.1026.1 Function

Quaestor-Event-Ended-At names an undocumented function, with lambda list (INSTANCE).

### 8.1026.2 File

Defined in file src/db/friendly.lisp.

### 8.1026.3 SetF Function

(SETF Quaestor-Event-Ended-At) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1026.4 File

Defined in file src/db/friendly.lisp.

## **8.1027 Tootsville::Quaestor-Event-Fairy-Dust**

### **8.1027.1 Function**

Quaestor-Event-Fairy-Dust names an undocumented function, with lambda list (INSTANCE).

### **8.1027.2 File**

Defined in file src/db/friendly.lisp.

### **8.1027.3 SetF Function**

(SETF Quaestor-Event-Fairy-Dust) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1027.4 File**

Defined in file src/db/friendly.lisp.

## 8.1028 Tootsville::Quaestor-Event-Item

### 8.1028.1 Function

Quaestor-Event-Item names an undocumented function, with lambda list (INSTANCE).

### 8.1028.2 File

Defined in file src/db/friendly.lisp.

### 8.1028.3 SetF Function

(SETF Quaestor-Event-Item) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1028.4 File

Defined in file src/db/friendly.lisp.

## **8.1029 Tootsville::Quaestor-Event-Medal**

### **8.1029.1 Function**

Quaestor-Event-Medal names an undocumented function, with lambda list (INSTANCE).

### **8.1029.2 File**

Defined in file src/db/friendly.lisp.

### **8.1029.3 SetF Function**

(SETF Quaestor-Event-Medal) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1029.4 File**

Defined in file src/db/friendly.lisp.



## **8.1030 Tootsville::Quaestor-Event-P**

### **8.1030.1 Function**

Quaestor-Event-P names an undocumented function, with lambda list (OBJECT).

### **8.1030.2 File**

Defined in file src/db/friendly.lisp.

## **8.1031 Tootsville::Quaestor-Event-Peanuts**

### **8.1031.1 Function**

Quaestor-Event-Peanuts names an undocumented function, with lambda list (INSTANCE).

### **8.1031.2 File**

Defined in file src/db/friendly.lisp.

### **8.1031.3 SetF Function**

(SETF Quaestor-Event-Peanuts) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1031.4 File**

Defined in file src/db/friendly.lisp.

## 8.1032 Tootsville::Quaestor-Event-Score

### 8.1032.1 Function

Quaestor-Event-Score names an undocumented function, with lambda list (INSTANCE).

### 8.1032.2 File

Defined in file src/db/friendly.lisp.

### 8.1032.3 SetF Function

(SETF Quaestor-Event-Score) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1032.4 File

Defined in file src/db/friendly.lisp.

## **8.1033 Tootsville::Quaestor-Event-Source**

### **8.1033.1 Function**

Quaestor-Event-Source names an undocumented function, with lambda list (INSTANCE).

### **8.1033.2 File**

Defined in file src/db/friendly.lisp.

### **8.1033.3 SetF Function**

(SETF Quaestor-Event-Source) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1033.4 File**

Defined in file src/db/friendly.lisp.

## 8.1034 Tootsville::Quaestor-Event-Started-At

### 8.1034.1 Function

Quaestor-Event-Started-At names an undocumented function, with lambda list (INSTANCE).

### 8.1034.2 File

Defined in file src/db/friendly.lisp.

### 8.1034.3 SetF Function

(SETF Quaestor-Event-Started-At) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1034.4 File

Defined in file src/db/friendly.lisp.

## **8.1035 Tootsville::Quaestor-Event-Started-By**

### **8.1035.1 Function**

Quaestor-Event-Started-By names an undocumented function, with lambda list (INSTANCE).

### **8.1035.2 File**

Defined in file src/db/friendly.lisp.

### **8.1035.3 SetF Function**

(SETF Quaestor-Event-Started-By) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1035.4 File**

Defined in file src/db/friendly.lisp.

## **8.1036 Tootsville::Quaestor-Event-Uuid**

### **8.1036.1 Function**

Quaestor-Event-Uuid names an undocumented function, with lambda list (INSTANCE).

### **8.1036.2 File**

Defined in file src/db/friendly.lisp.

### **8.1036.3 SetF Function**

(SETF Quaestor-Event-Uuid) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1036.4 File**

Defined in file src/db/friendly.lisp.

## **8.1037 Tootsville::Quaestor-New-Toot**

### **8.1037.1 Function**

Quaestor-New-Toot names a function, with lambda list (TOOT):

Give the new TOOT their starting peanuts.

### **8.1037.2 File**

Defined in file src/quaestor.lisp.



## 8.1038 Tootsville::Quaestor-Start-Event

### 8.1038.1 Function

Quaestor-Start-Event names a function, with lambda list (MONIKER &OPTIONAL (TOOT \*TOOT\*)):

TOOT wants to start an event identified by MONIKER.

See Section 8.738 [TOOTSVILLE INFINITY-START-EVENT], page 1020, for details of the procedure.

### 8.1038.2 File

Defined in file src/quaestor.lisp.

## **8.1039 Tootsville::Quaestor-Start-General**

### **8.1039.1 Function**

Quaestor-Start-General names a function, with lambda list (ITEM TOOT):

Start a general event sourced on ITEM for TOOT.

### **8.1039.2 File**

Defined in file src/quaestor.lisp.

## **8.1040 Tootsville::Query-Params**

### **8.1040.1 Function**

Query-Params names a function, with lambda list NIL:

Get parameters from the query string of the current Hunchentoot request.

### **8.1040.2 File**

Defined in file src/web.lisp.

## **8.1041 Tootsville::Query-String->Plist**

### **8.1041.1 Function**

Query-String->Plist names a function, with lambda list (QUERY-STRING):

Split an HTTP QUERY-STRING into a PList.

XXX Probably a duplicate of something done in Hunchentoot or Drakma?

### **8.1041.2 File**

Defined in file src/web.lisp.

## 8.1042 Tootsville::Query-To-Memcache-Key

### 8.1042.1 Function

Query-To-Memcache-Key names a function, with lambda list (DB PREPARED ARGS):

Creates a key based on DB, PREPARED statement, and ARGS suitable for Mem-CacheD.

Currently uses Section 8.1131 [TOOTSVILLE SHA1-HEX], page 1416, of a particular stringified form

### 8.1042.2 File

Defined in file src/db/memcached.lisp.

## **8.1043 Tootsville::Quiesce-Connected-Toots**

### **8.1043.1 Function**

Quiesce-Connected-Toots names a function, with lambda list NIL:

Send every Toot a demand that it quiesce itself to the database.

FIXME: IGNORE-NOT-FOUND is because ... um ... Superstar sucks.

See: Section 8.360 [TOOTSVILLE DEMAND-QUIESCE-TOOT], page 616,  
Section 8.254 [TOOTSVILLE CONNECTED-TOOTS], page 508,

### **8.1043.2 File**

Defined in file src/toots.lisp.

## **8.1044 Tootsville::Rad-Personality**

### **8.1044.1 Class**

Rad-Personality names a class, with one superclass: Section 8.1097 [TOOTSVILLE ROBOT-RAD], page 1382.

This class defines a character named Rad

### **8.1044.2 Slots**

Class Rad-Personality has no direct slots defined.

## **8.1045 Tootsville::Random-Key**

### **8.1045.1 Function**

Random-Key names an undocumented function, with lambda list (OBJECT).

### **8.1045.2 SetF Function**

(SETF Random-Key) names an undocumented function, with lambda list (NEW-VALUE OBJECT).



## 8.1046 Tootsville::Random-Start-Wtl-For-Toot

### 8.1046.1 Function

Random-Start-Wtl-For-Toot names a function, with lambda list NIL:

Designate a starting position in Toot Square for a Toot.

Returns a WTL-type structure in a JSON string, with `course` and `facing` values.

Starting positions are randomly dispersed around the Toot Square fountain, which intentionally is the center of the coordinate system of the world.

### 8.1046.2 File

Defined in file `src/infinity/new-commands-20.lisp`.

## **8.1047 Tootsville::Raw-Post-String**

### **8.1047.1 Function**

Raw-Post-String names a function, with lambda list NIL:

Obtain POSTed data as a string

### **8.1047.2 File**

Defined in file src/web.lisp.

## 8.1048 Tootsville::Read-Related-Journal

### 8.1048.1 Function

Read-Related-Journal names a function, with lambda list (WHO &KEY LAST):

Read staff journal entries related to WHO.

Or, read the single item LAST from the end; when LAST = 0, the very latest entry; when LAST < 0, then the “nth” entry from the end. Thus, -1 is the next-to-last entry, -2 is the third from the end.

WHO may be anything accepted by Section 8.544 [TOOTSVILLE ENSURE-LIST-OF-PEOPLE], page 801.

### 8.1048.2 File

Defined in file src/staff-journal.lisp.

## 8.1049 Tootsville::Read-Staff-Journal

### 8.1049.1 Function

Read-Staff-Journal names a function, with lambda list (&KEY (START-DATE (YESTERDAY)) (END-DATE (NOW)) LAST):

Read staff journal entries between (inclusive) START-DATE and END-DATE; default, yesterday and today.

Or, read the single item LAST from the end; when LAST = 0, the very latest entry; when LAST < 0, then the “nth” entry from the end. Thus, -1 is the next-to-last entry, -2 is the third from the end.

### 8.1049.2 File

Defined in file src/staff-journal.lisp.

## **8.1050 Tootsville::Reap-Uninteresting-Child-Requests**

### **8.1050.1 Function**

Reap-Uninteresting-Child-Requests names a function, with lambda list NIL:

Remove uninteresting requests from the `child_requests` table.

Normally run by the metronome periodically.

### **8.1050.2 File**

Defined in file `src/users.lisp`.

## **8.1051 Tootsville::Reasonable-Name-Char-P**

### **8.1051.1 Function**

Reasonable-Name-Char-P names a function, with lambda list (CHAR):

Is CHAR a character that can reasonably appear in a person's name?

### **8.1051.2 File**

Defined in file src/users.lisp.

## **8.1052 Tootsville::Reasonable-Name-P**

### **8.1052.1 Function**

Reasonable-Name-P names a function, with lambda list (NAME):

Does NAME appear to be a reasonable name for a person?

### **8.1052.2 File**

Defined in file src/users.lisp.

## **8.1053 Tootsville::Rebuild-Myself**

### **8.1053.1 Function**

Rebuild-Myself names a function, with lambda list NIL:

Recompile the running server.

Hopefully you've already tested the changes?

### **8.1053.2 File**

Defined in file src/main.lisp.



## 8.1054 Tootsville::Redirect-To

### 8.1054.1 Function

Redirect-To names a function, with lambda list (URI &OPTIONAL (STATUS 307)):

Redirect to another URI. Status code 307 for temporary, 301 or 308 for permanent (typically). (:TEMPORARY and :PERMANENT are accepted for readability.)

As a side effect, provides an extremely skeletal HTML redirection page via 'REDIRECT-TO/HTML/BODY'.

### 8.1054.2 File

Defined in file src/redirect.lisp.

## **8.1055 Tootsville::Redirect-To/ Html-Body**

### **8.1055.1 Function**

Redirect-To/ Html-Body names a function, with lambda list (URI):

Returns an octet array that gives a simple redirection link.

This is a silly legacy thing for ancient browsers that don't follow a 3xx redirection or want to display something while they're redirecting. In real life, it's rarely encountered by a real browser, but sometimes caught by tools like curl or wget with certain settings.

### **8.1055.2 File**

Defined in file src/redirect.lisp.

## 8.1056 Tootsville::Register-Metronome-Tasks

### 8.1056.1 Function

Register-Metronome-Tasks names a function, with lambda list NIL:

Register certain metronome tasks for miscellaneous services.

This is a list of specific facilities that are started up during the system boot process.

#### Websocket AYT facility

This facility, Section 8.150 [TOOTSVILLE AYT-IDLE-USERS], page 404, runs every 120 seconds to detect and disconnect users who are no longer actually connected. (Note that AYT is netspeak for “are you there?”)

#### Toot Quiesce facility

This facility runs every 600 seconds to asks Toots to quiesce themselves to the database. See Section 8.1043 [TOOTSVILLE QUIESCE-CONNECTED-TOOTS], page 1328.

#### Reap uninteresting child requests

See Section 8.1050 [TOOTSVILLE REAP-UNINTERESTING-CHILD-REQUESTS], page 1335. Every 4 hours clears out some uninteresting records from the “child\_requests” database table.

### 8.1056.2 File

Defined in file src/metronome.lisp.

## **8.1057 Tootsville::Register-Signal-Handlers**

### **8.1057.1 Function**

Register-Signal-Handlers names an undocumented function, with lambda list NIL.

### **8.1057.2 File**

Defined in file src/main.lisp.

## **8.1058 Tootsville::Relative-Facing**

### **8.1058.1 Function**

Relative-Facing names an undocumented function, with lambda list (X1 Z1 X2 Z2).

### **8.1058.2 File**

Defined in file `src/characters/robots.lisp`.

## **8.1059 Tootsville::Reload-Production**

### **8.1059.1 Function**

Reload-Production names an undocumented function, with lambda list NIL.

### **8.1059.2 File**

Defined in file src/main.lisp.

## **8.1060 Tootsville::Remap-Endpoints**

### **8.1060.1 Function**

Remap-Endpoints names an undocumented function, with lambda list NIL.

### **8.1060.2 File**

Defined in file src/endpoint.lisp.

## **8.1061 Tootsville::Remove-Furniture**

### **8.1061.1 Function**

Remove-Furniture names an undocumented function, with lambda list (SLOT).

### **8.1061.2 File**

Defined in file `src/infinity/legacy-commands.lisp`.



## **8.1062 Tootsville::Remove-Repeats-For-Toot-Name**

### **8.1062.1 Function**

Remove-Repeats-For-Toot-Name names a function, with lambda list (STRING):

Remove repeated characters from STRING.

Removes letters that repeat more than twice in a row, or hyphens that occur more than once in a row.

### **8.1062.2 File**

Defined in file src/types/toot-names.lisp.

## **8.1063 Tootsville::Rename-Toot**

### **8.1063.1 Function**

Rename-Toot names a function, with lambda list (TOOT NEW-NAME):

Rename TOOT to NEW-NAME.

### **8.1063.2 File**

Defined in file src/toots.lisp.

## **8.1064 Tootsville::Render-Json**

### **8.1064.1 Function**

Render-Json names an undocumented function, with lambda list (OBJECT).

### **8.1064.2 File**

Defined in file src/view.lisp.

## **8.1065 Tootsville::Replace-TeXinfo-Tables**

### **8.1065.1 Function**

Replace-TeXinfo-Tables names an undocumented function, with lambda list (STRING).

### **8.1065.2 File**

Defined in file src/endpoints/slash-meta-game.lisp.

## **8.1066 Tootsville::Report-Slow-Query**

### **8.1066.1 Function**

Report-Slow-Query names an undocumented function, with lambda list (FNAME ELAPSED HOW-SLOW-IS-SLOW).

### **8.1066.2 File**

Defined in file src/web.lisp.

## **8.1067 Tootsville::Request-Accept-Types**

### **8.1067.1 Function**

Request-Accept-Types names a function, with lambda list NIL:

Determine the Accept: types from the current HTTP request headers.

### **8.1067.2 File**

Defined in file src/acceptor.lisp.

## **8.1068 Tootsville::Respond-To-Error**

### **8.1068.1 Function**

Respond-To-Error names an undocumented function, with lambda list (CONDITION).

### **8.1068.2 File**

Defined in file src/acceptor.lisp.

## **8.1069 Tootsville::Restore-Robot-Wtl**

### **8.1069.1 Function**

Restore-Robot-Wtl names a function, with lambda list (ROBOT):

Restore the walk-the-line positioning data for ROBOT

Pulls quiesced data, where available, or creates a new one with Section 8.1046 [TOOTSVILLE RANDOM-START-WTL-FOR-TOOT], page 1331, if no quiescent data is available.

### **8.1069.2 File**

Defined in file src/characters/robots.lisp.



## 8.1070 Tootsville::Return-New-Apple

### 8.1070.1 Function

Return-New-Apple names a function, with lambda list (CLIENT):

Used by Section 8.693 [TOOTSVILLE INFINITY-GET-APPLE], page 956, to send CLIENT a new apple value.

### 8.1070.2 File

Defined in file src/websockets.lisp.

## 8.1071 Tootsville::Rgb-Bytes->Rgb

### 8.1071.1 Function

Rgb-Bytes->Rgb names a function, with lambda list (BYTES):

Convert BYTES into a list of red, green, and blue values.

BYTES is an RGB triplet of 3 8-bit bytes, like Section 8.233 [TOOTSVILLE COLOR24-TO-INTEGER], page 487, or Section 8.749 [TOOTSVILLE INTEGER-TO-COLOR24], page 1034, representation; i.e. an integer 24 bits long of which the upper 8 bits are the red channel, next 8 bits are green, and lower 8 bits are the blue channel.

### 8.1071.2 File

Defined in file src/types/color+pattern.lisp.

## **8.1072 Tootsville::Robot**

### **8.1072.1 Class**

Robot names a class, with one superclass: COMMON-LISP::STANDARD-OBJECT (not in this manual).

An in-game robot character

### **8.1072.2 Slots**

Class Robot has no direct slots defined.

## **8.1073 Tootsville::Robot-Broadcast**

### **8.1073.1 Function**

Robot-Broadcast names a function, with lambda list (MESSAGE NEAR &KEY EXCEPT):  
Broadcast MESSAGE to all robots near NEAR, except robot EXCEPT.

### **8.1073.2 File**

Defined in file src/characters/robots.lisp.

## 8.1074 Tootsville::Robot-Chaos

### 8.1074.1 Class

Robot-Chaos names a class, with one superclass: Section 8.1132 [TOOTSVILLE SHADOW-PERSONALITY], page 1417.

### 8.1074.2 Slots

Class Robot-Chaos has no direct slots defined.

## **8.1075 Tootsville::Robot-Course**

### **8.1075.1 Function**

Robot-Course names an undocumented function, with lambda list (OBJECT).

### **8.1075.2 SetF Function**

(SETF Robot-Course) names an undocumented function, with lambda list (NEW-VALUE OBJECT).

## **8.1076 Tootsville::Robot-Course-Wtl**

### **8.1076.1 Function**

Robot-Course-Wtl names a function, with lambda list (ROBOT):

Get the course of ROBOT in Walk-The-Line JSON form.

### **8.1076.2 File**

Defined in file src/characters/robots.lisp.

## **8.1077 Tootsville::Robot-Cupid**

### **8.1077.1 Class**

Robot-Cupid names a class, with one superclass: Section 8.163 [TOOTSVILLE BASIC-8-PERSONALITY], page 417.

### **8.1077.2 Slots**

Class Robot-Cupid has no direct slots defined.



## **8.1078 Tootsville::Robot-Doodle**

### **8.1078.1 Class**

Robot-Doodle names a class, with one superclass: Section 8.1269 [TOOTSVILLE TOOT-PERSONALITY], page 1556.

### **8.1078.2 Slots**

Class Robot-Doodle has no direct slots defined.

## **8.1079 Tootsville::Robot-Dottie**

### **8.1079.1 Class**

Robot-Dottie names a class, with one superclass: Section 8.163 [TOOTSVILLE BASIC-8-PERSONALITY], page 417.

### **8.1079.2 Slots**

Class Robot-Dottie has no direct slots defined.

## **8.1080 Tootsville::Robot-Flora**

### **8.1080.1 Class**

Robot-Flora names a class, with one superclass: Section 8.163 [TOOTSVILLE BASIC-8-PERSONALITY], page 417.

### **8.1080.2 Slots**

Class Robot-Flora has no direct slots defined.

## 8.1081 Tootsville::Robot-Go-To

### 8.1081.1 Function

Robot-Go-To names an undocumented function, with lambda list (ROBOT X Y Z &OPTIONAL (SPEED)).

## 8.1082 Tootsville::Robot-Handle

### 8.1082.1 Function

Robot-Handle names a function, with lambda list (ROBOT FROM STATUS MESSAGE):

Called for ROBOT to handle a MESSAGE with from-tag FROM and status-tag STATUS.

Infinity protocol message MESSAGE was received by the ROBOT. It has been parsed from JSON form into a plist, and the `from` and `status` tags have been broken out, with the `from` element converted into a keyword argument.

Methods on this generic function will likely specialize on ROBOT by class, and FROM & STATUS using EQL.

### 8.1082.2 File

Defined in file `src/characters/robo-toot.lisp`.

## **8.1083 Tootsville::Robot-Harmony**

### **8.1083.1 Class**

Robot-Harmony names a class, with one superclass: Section 8.1269 [TOOTSVILLE TOOT-PERSONALITY], page 1556.

### **8.1083.2 Slots**

Class Robot-Harmony has no direct slots defined.

## **8.1084 Tootsville::Robot-Has-Heard**

### **8.1084.1 Function**

Robot-Has-Heard names an undocumented function, with lambda list (OBJECT).

### **8.1084.2 SetF Function**

(SETF Robot-Has-Heard) names an undocumented function, with lambda list (NEW-VALUE OBJECT).

## 8.1085 Tootsville::Robot-Heard

### 8.1085.1 Function

Robot-Heard names a function, with lambda list (ROBOT SPEAKER MODE HEARD):

Robot ROBOT heard SPEAKER in mode MODE say HEARD.

SPEAKER is a Toot character. MODE is the mode in which ROBOT has placed its conversation with SPEAKER, and is typically a keyword, but defaults to NIL. HEARD is an array of the most recent utterances from SPEAKER, in the order in which they were received; thus, the latest utterance is (LASTCAR HEARD).

### 8.1085.2 File

Defined in file src/characters/robo-toot.lisp.



## **8.1086 Tootsville::Robot-Jack**

### **8.1086.1 Class**

Robot-Jack names a class, with one superclass: Section 8.663 [TOOTSVILLE HOLIDAY-SPECIAL-PERSONALITY], page 921.

### **8.1086.2 Slots**

Class Robot-Jack has no direct slots defined.

## **8.1087 Tootsville::Robot-Lil-Mc**

### **8.1087.1 Class**

Robot-Lil-Mc names a class, with one superclass: Section 8.163 [TOOTSVILLE BASIC-8-PERSONALITY], page 417.

### **8.1087.2 Slots**

Class Robot-Lil-Mc has no direct slots defined.

## 8.1088 Tootsville::Robot-Listen

### 8.1088.1 Function

Robot-Listen names a function, with lambda list (ROBOT LISTENER-NAME SPEAKER TEXT VOLUME):

ROBOT, named LISTENER-NAME, heard SPEAKER say TEXT with volume VOLUME.

You probably mean to specialize Section 8.1085 [TOOTSVILLE ROBOT-HEARD], page 1370, q.v. This method calls ROBOT-HEARD in turn.

### 8.1088.2 File

Defined in file src/characters/robo-toot.lisp.

## **8.1089 Tootsville::Robot-Match**

### **8.1089.1 Macro**

Robot-Match names an undocumented macro, with lambda list ((&REST STRINGS) &BODY BODY).

### **8.1089.2 File**

Defined in file src/characters/robots.lisp.

## **8.1090 Tootsville::Robot-Mayor-Louis**

### **8.1090.1 Class**

Robot-Mayor-Louis names a class, with one superclass: Section 8.1132 [TOOTSVILLE SHADOW-PERSONALITY], page 1417.

### **8.1090.2 Slots**

Class Robot-Mayor-Louis has no direct slots defined.

## **8.1091 Tootsville::Robot-Mode**

### **8.1091.1 Function**

Robot-Mode names an undocumented function, with lambda list (OBJECT).

### **8.1091.2 SetF Function**

(SETF Robot-Mode) names an undocumented function, with lambda list (NEW-VALUE OBJECT).

## **8.1092 Tootsville::Robot-Moo**

### **8.1092.1 Class**

Robot-Moo names a class, with one superclass: Section 8.163 [TOOTSVILLE BASIC-8-PERSONALITY], page 417.

### **8.1092.2 Slots**

Class Robot-Moo has no direct slots defined.

## **8.1093 Tootsville::Robot-Nevermind**

### **8.1093.1 Class**

Robot-Nevermind names a class, with one superclass: Section 8.1132 [TOOTSVILLE SHADOW-PERSONALITY], page 1417.

### **8.1093.2 Slots**

Class Robot-Nevermind has no direct slots defined.



## **8.1094 Tootsville::Robot-Picasso**

### **8.1094.1 Class**

Robot-Picasso names a class, with one superclass: Section 8.1269 [TOOTSVILLE TOOT-PERSONALITY], page 1556.

### **8.1094.2 Slots**

Class Robot-Picasso has no direct slots defined.

## **8.1095 Tootsville::Robot-Position**

### **8.1095.1 Function**

Robot-Position names an undocumented function, with lambda list (ROBOT).

### **8.1095.2 File**

Defined in file `src/characters/robots.lisp`.

## **8.1096 Tootsville::Robot-Props**

### **8.1096.1 Class**

Robot-Props names a class, with one superclass: Section 8.1269 [TOOTSVILLE TOOT-PERSONALITY], page 1556.

### **8.1096.2 Slots**

Class Robot-Props has no direct slots defined.

## **8.1097 Tootsville::Robot-Rad**

### **8.1097.1 Class**

Robot-Rad names a class, with one superclass: Section 8.1269 [TOOTSVILLE TOOTPERSONALITY], page 1556.

### **8.1097.2 Slots**

Class Robot-Rad has no direct slots defined.

## **8.1098 Tootsville::Robot-Say**

### **8.1098.1 Function**

Robot-Say names an undocumented function, with lambda list (ROBOT FORMAT &REST FORMAT-ARGS).

## **8.1099 Tootsville::Robot-Set-Mode**

### **8.1099.1 Macro**

Robot-Set-Mode names an undocumented macro, with lambda list (MODE).

### **8.1099.2 File**

Defined in file `src/characters/robots.lisp`.

## **8.1100 Tootsville::Robot-Shade**

### **8.1100.1 Class**

Robot-Shade names a class, with one superclass: Section 8.1072 [TOOTSVILLE ROBOT], page 1357.

### **8.1100.2 Slots**

Class Robot-Shade has no direct slots defined.

## **8.1101 Tootsville::Robot-Smudge**

### **8.1101.1 Class**

Robot-Smudge names a class, with one superclass: Section 8.1132 [TOOTSVILLE SHADOW-PERSONALITY], page 1417.

### **8.1101.2 Slots**

Class Robot-Smudge has no direct slots defined.



## 8.1102 Tootsville::Robot-Snowcone

### 8.1102.1 Class

Robot-Snowcone names a class, with one superclass: Section 8.663 [TOOTSVILLE HOLIDAY-SPECIAL-PERSONALITY], page 921.

### 8.1102.2 Slots

Class Robot-Snowcone has no direct slots defined.

## **8.1103 Tootsville::Robot-Sparkle**

### **8.1103.1 Class**

Robot-Sparkle names a class, with one superclass: Section 8.163 [TOOTSVILLE BASIC-8-PERSONALITY], page 417.

### **8.1103.2 Slots**

Class Robot-Sparkle has no direct slots defined.

## 8.1104 Tootsville::Robot-Sploit

### 8.1104.1 Class

Robot-Sploit names a class, with one superclass: Section 8.1132 [TOOTSVILLE SHADOW-PERSONALITY], page 1417.

### 8.1104.2 Slots

Class Robot-Sploit has no direct slots defined.

## **8.1105 Tootsville::Robot-Superstar**

### **8.1105.1 Class**

Robot-Superstar names a class, with one superclass: Section 8.163 [TOOTSVILLE BASIC-8-PERSONALITY], page 417.

### **8.1105.2 Slots**

Class Robot-Superstar has no direct slots defined.

## 8.1106 Tootsville::Robot-Unicast

### 8.1106.1 Function

Robot-Unicast names a function, with lambda list (MESSAGE ROBOT):

Send MESSAGE to ROBOT only.

MESSAGE is a JSON-encoded string, or a plist approximating one. ROBOT is a robot or Toot object.

### 8.1106.2 File

Defined in file src/characters/robo-toot.lisp.

## **8.1107 Tootsville::Robot-Zap**

### **8.1107.1 Class**

Robot-Zap names a class, with one superclass: Section 8.163 [TOOTSVILLE BASIC-8-PERSONALITY], page 417.

### **8.1107.2 Slots**

Class Robot-Zap has no direct slots defined.

## **8.1108 Tootsville::Robotp**

### **8.1108.1 Function**

Robotp names a function, with lambda list (USER):

Is USER a robot?

USER may be a robot or a Toot that is controlled by a robot.

### **8.1108.2 File**

Defined in file src/characters/robots.lisp.

## **8.1109 Tootsville::Romance-Ii-Copyright-Latest**

### **8.1109.1 Function**

Romance-Ii-Copyright-Latest names an undocumented function, with lambda list NIL.

### **8.1109.2 File**

Defined in file src/version.lisp.



## **8.1110 Tootsville::Romance-Ii-Program-Name**

### **8.1110.1 Function**

Romance-Ii-Program-Name names a function, with lambda list NIL:

This program's name. Taken from ASDF.

### **8.1110.2 File**

Defined in file src/version.lisp.

## **8.1111 Tootsville::Romance-Ii-Program-Name/ Version**

### **8.1111.1 Function**

Romance-Ii-Program-Name/ Version names a function, with lambda list NIL:

This program's name and version number, in name/version form, as used in HTTP headers and such.

### **8.1111.2 File**

Defined in file src/version.lisp.

## **8.1112 Tootsville::Romance-Ii-Program-Version**

### **8.1112.1 Function**

Romance-Ii-Program-Version names a function, with lambda list NIL:

This program's version. Taken from ASDF.

### **8.1112.2 File**

Defined in file src/version.lisp.

## **8.1113 Tootsville::Run-Async**

### **8.1113.1 Function**

Run-Async names a function, with lambda list (FUNCTION &KEY NAME):

Run FUNCTION asynchronously in a thread named NAME.

If NAME is omitted, a generic name will be created based on FUNCTION.

### **8.1113.2 File**

Defined in file src/main.lisp.

## 8.1114 Tootsville::Run-Metronome-Tasks

### 8.1114.1 Function

Run-Metronome-Tasks names a function, with lambda list NIL:

Runs tasks scheduled for the game’s metronome.

Typically these tasks are scheduled in one of three ways. They may be scheduled to occur at a given frequency in seconds, at a single time, or at a give frequency up until a certain time.

Tasks are usually created by Section 8.379 [TOOTSVILLE DO-METRONOME], page 635, which in turn uses Section 8.896 [TOOTSVILLE METRONOME-REGISTER], page 1181, to safely enqueue the tasks with locking.

The metronome runs at approximately 1 second resolution, but steps its time forward at precisely 1 second intervals, so no task will be missed due to system scheduler tie-ups.

Tasks are not allowed to “stack up;” if a task has not finished by the time its next execution window comes around, it will miss its opportunity and have to wait for the next window.

### 8.1114.2 File

Defined in file src/metronome.lisp.

## 8.1115 Tootsville::Save-Record

### 8.1115.1 Function

Save-Record names a function, with lambda list (OBJECT):

Write OBJECT to the database, with any changes made.

Types are encouraged to introduce appropriate consistency checks into a :BEFORE method on this function. The default :AFTER method calls Section 8.751 [TOOTSVILLE INVALIDATE-CACHE], page 1036,

### 8.1115.2 File

Defined in file src/db/generic-db.lisp.

## **8.1116 Tootsville::Send-Parent-Child-Login-Email**

### **8.1116.1 Function**

Send-Parent-Child-Login-Email names a function, with lambda list (REQUEST):

Send a parent child's REQUEST to play via email.

### **8.1116.2 File**

Defined in file src/users.lisp.

## **8.1117 Tootsville::Send-Parent-Child-Login-Request**

### **8.1117.1 Function**

Send-Parent-Child-Login-Request names a function, with lambda list (REQUEST):

Send a parent a child's REQUEST to play as a popup in game.

### **8.1117.2 File**

Defined in file src/users.lisp.



## **8.1118 Tootsville::Send-Reply-As-Bytes**

### **8.1118.1 Function**

Send-Reply-As-Bytes names an undocumented function, with lambda list (REPLY FN-AME).

### **8.1118.2 File**

Defined in file src/web.lisp.

## 8.1119 Tootsville::Send-Sms-Message

### 8.1119.1 Function

Send-Sms-Message names a function, with lambda list (&KEY FROM TO BODY UUID):

Send the SMS message from FROM to TO with body BODY. On success or error, reference UUID.

If online, both FROM and TO will receive notifications.

FROM and TO may be Toot designators, or TO may be a list of Toot designators.

See Section 8.727 [TOOTSVILLE INFINITY-SEND-MAIL-MESSAGE], page 1004.

### 8.1119.2 File

Defined in file src/sms.lisp.

## **8.1120 Tootsville::Server-List**

### **8.1120.1 Function**

Server-List names a function, with lambda list NIL:

A list of all servers active in the current cluster.

### **8.1120.2 File**

Defined in file src/tcp-stream.lisp.

## **8.1121 Tootsville::Set-Http-Default-Headers**

### **8.1121.1 Function**

Set-Http-Default-Headers names an undocumented function, with lambda list NIL.

### **8.1121.2 File**

Defined in file src/acceptor.lisp.

## **8.1122 Tootsville::Set-Up-For-Daemon/ Error-Output**

### **8.1122.1 Function**

Set-Up-For-Daemon/ Error-Output names a function, with lambda list (LOG-DIR):

Set up the \*ERROR-OUTPUT\* (see the Common Lisp HyperSpec) for logging in LOG-DIR.

### **8.1122.2 File**

Defined in file src/logging.lisp.

## **8.1123 Tootsville::Set-Up-For-Daemon/ Log-Output**

### **8.1123.1 Function**

Set-Up-For-Daemon/ Log-Output names a function, with lambda list (LOG-DIR):

Set up the Verbose mode logging output file in LOG-DIR.

### **8.1123.2 File**

Defined in file src/logging.lisp.

## **8.1124 Tootsville::Set-Up-For-Daemon/ Standard-Output**

### **8.1124.1 Function**

Set-Up-For-Daemon/ Standard-Output names a function, with lambda list (LOG-DIR):

Set up the `*STANDARD-OUTPUT*` (see the Common Lisp HyperSpec) for logging.

### **8.1124.2 File**

Defined in file `src/logging.lisp`.

## **8.1125 Tootsville::Set-Up-For-Daemon/ Start-Logging**

### **8.1125.1 Function**

Set-Up-For-Daemon/ Start-Logging names a function, with lambda list NIL:

Set up for daemon-mode logging.

### **8.1125.2 File**

Defined in file src/logging.lisp.



## **8.1126 Tootsville::Set-Up-For-Daemon/ Trace-Output**

### **8.1126.1 Function**

Set-Up-For-Daemon/ Trace-Output names a function, with lambda list (LOG-DIR):

Set up the \*TRACE-OUTPUT\* (see the Common Lisp HyperSpec) for logging in LOG-DIR.

### **8.1126.2 File**

Defined in file src/logging.lisp.

## **8.1127 Tootsville::Set-User-Var**

### **8.1127.1 Function**

Set-User-Var names an undocumented function, with lambda list (TOOT KEY VALUE).

### **8.1127.2 File**

Defined in file src/infinity/legacy-commands.lisp.

## **8.1128 Tootsville::Set-User-Var-D**

### **8.1128.1 Function**

Set-User-Var-D names an undocumented function, with lambda list (TOOT VALUE).

### **8.1128.2 File**

Defined in file `src/infinity/legacy-commands.lisp`.

## **8.1129 Tootsville::Set-User-Var-Wtl**

### **8.1129.1 Function**

Set-User-Var-Wtl names an undocumented function, with lambda list (TOOT VALUE).

### **8.1129.2 File**

Defined in file src/infinity/legacy-commands.lisp.

## 8.1130 Tootsville::Sha1-Hash

### 8.1130.1 Function

Sha1-Hash names a function, with lambda list (MESSAGE):

Get the hex-string hash of MESSAGE, which is an UTF-8 string.

### 8.1130.2 File

Defined in file `src/endpoints/gossip/alex/alex.lisp`.

## **8.1131 Tootsville::Sha1-Hex**

### **8.1131.1 Function**

Sha1-Hex names a function, with lambda list (STRING):

Compute the SHA1 hash of STRING and return it as a string of hex digits.

### **8.1131.2 File**

Defined in file src/types/binary.lisp.

## **8.1132 Tootsville::Shaddow-Personality**

### **8.1132.1 Class**

Shaddow-Personality names a class, with one superclass: Section 8.1072 [TOOTSVILLE ROBOT], page 1357.

### **8.1132.2 Slots**

Class Shaddow-Personality has no direct slots defined.

## **8.1133 Tootsville::Shade-Personality**

### **8.1133.1 Class**

Shade-Personality names a class, with one superclass: Section 8.1100 [TOOTSVILLE ROBOT-SHADE], page 1385.

This class defines a character named Shade

### **8.1133.2 Slots**

Class Shade-Personality has no direct slots defined.



## 8.1134 Tootsville::Shift-Contour-Point

### 8.1134.1 Function

Shift-Contour-Point names a function, with lambda list (LATITUDE LONGITUDE SHIFT):

Shift a point on the contour map vertically

### 8.1134.2 File

Defined in file src/terrain.lisp.

## **8.1135 Tootsville::Sinus**

### **8.1135.1 Function**

Sinus names a function, with lambda list (X RANGE):

Give the Y value at X in a sinus curve

### **8.1135.2 File**

Defined in file src/utls.lisp.

## 8.1136 Tootsville::Sky-Contents

### 8.1136.1 Function

Sky-Contents names an undocumented function, with lambda list (X Y Z &OPTIONAL (NOW (GET-UNIVERSAL-TIME))).

### 8.1136.2 File

Defined in file src/world.lisp.

## 8.1137 Tootsville::Sky-Room-Var

### 8.1137.1 Function

Sky-Room-Var names a function, with lambda list (WORLD):

Returns the current state of the skies over WORLD.

This data is in the form of a Plist suitable for JSON-ification. It's expected to be used by Section 8.819 [TOOTSVILLE LOCAL-ROOM-VARS], page 1104, particularly, q.v.

When WORLD is CHOR (Chœrogyllum), the sky will contain a sun, and three moons. For each body, the X and Y positions will be returned; in addition, for each moon, the phase ( $\phi$ ) of the moon will be returned.

### 8.1137.2 Example structure

```
{ sun: { x: 120, y: 120 },  
  moon: { x: 120, y: 120,  $\phi$ : 1 },  
  othM: { x: 120, y: 120,  $\phi$ : 1 },  
  pink: { x: 120, y: 120,  $\phi$ : 1 } }
```

### 8.1137.3 File

Defined in file src/infinity/new-commands-20.lisp.

## 8.1138 Tootsville::Slot-Values

### 8.1138.1 Function

Slot-Values names a function, with lambda list (OBJECT):

For any OBJECT, this returns a list; each element is a PList with a slot name and value, encoded in JSON.

### 8.1138.2 File

Defined in file src/errors.lisp.

## **8.1139 Tootsville::Smoothe-Contour-200×200**

### **8.1139.1 Function**

Smoothe-Contour-200×200 names an undocumented function, with lambda list (LATITUDE LONGITUDE &OPTIONAL (REPEATS 3)).

### **8.1139.2 File**

Defined in file src/terrain.lisp.

## 8.1140 Tootsville::Sms

### 8.1140.1 Class

Sms names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.1140.2 Slots

Class Sms has 5 direct slot definitions:

Uuid

Sender

Destination

Message

Mmsp

## **8.1141 Tootsville::Sms-Destination**

### **8.1141.1 Function**

Sms-Destination names an undocumented function, with lambda list (INSTANCE).

### **8.1141.2 File**

Defined in file src/db/friendly.lisp.

### **8.1141.3 SetF Function**

(SETF Sms-Destination) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1141.4 File**

Defined in file src/db/friendly.lisp.



## 8.1142 Tootsville::Sms-Message

### 8.1142.1 Function

Sms-Message names an undocumented function, with lambda list (INSTANCE).

### 8.1142.2 File

Defined in file src/db/friendly.lisp.

### 8.1142.3 SetF Function

(SETF Sms-Message) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1142.4 File

Defined in file src/db/friendly.lisp.

## **8.1143 Tootsville::Sms-Message-Index**

### **8.1143.1 Function**

Sms-Message-Index names a function, with lambda list (TOOT UUID):

Find the SQL position of UUID in a TOOT's mailbox

### **8.1143.2 File**

Defined in file src/sms.lisp.

## **8.1144 Tootsville::Sms-Mmsp**

### **8.1144.1 Function**

Sms-Mmsp names an undocumented function, with lambda list (INSTANCE).

### **8.1144.2 File**

Defined in file src/db/friendly.lisp.

### **8.1144.3 SetF Function**

(SETF Sms-Mmsp) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1144.4 File**

Defined in file src/db/friendly.lisp.

## **8.1145 Tootsville::Sms-P**

### **8.1145.1 Function**

Sms-P names an undocumented function, with lambda list (OBJECT).

### **8.1145.2 File**

Defined in file src/db/friendly.lisp.

## 8.1146 Tootsville::Sms-Sender

### 8.1146.1 Function

Sms-Sender names an undocumented function, with lambda list (INSTANCE).

### 8.1146.2 File

Defined in file src/db/friendly.lisp.

### 8.1146.3 SetF Function

(SETF Sms-Sender) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1146.4 File

Defined in file src/db/friendly.lisp.

## **8.1147 Tootsville::Sms-Uuid**

### **8.1147.1 Function**

Sms-Uuid names an undocumented function, with lambda list (INSTANCE).

### **8.1147.2 File**

Defined in file src/db/friendly.lisp.

### **8.1147.3 SetF Function**

(SETF Sms-Uuid) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1147.4 File**

Defined in file src/db/friendly.lisp.

## 8.1148 Tootsville::Smudge-Personality

### 8.1148.1 Class

Smudge-Personality names a class, with one superclass: Section 8.1101 [TOOTSVILLE ROBOT-SMUDGE], page 1386.

This class defines a character named Smudge

### 8.1148.2 Slots

Class Smudge-Personality has no direct slots defined.

## **8.1149 Tootsville::Snowcone-Personality**

### **8.1149.1 Class**

Snowcone-Personality names a class, with one superclass: Section 8.1102 [TOOTSVILLE ROBOT-SNOWCONE], page 1387.

This class defines a character named Snowcone

### **8.1149.2 Slots**

Class Snowcone-Personality has no direct slots defined.



## 8.1150 Tootsville::Sparkle-Personality

### 8.1150.1 Class

Sparkle-Personality names a class, with one superclass: Section 8.1103 [TOOTSVILLE ROBOT-SPARKLE], page 1388.

This class defines a character named Sparkle

### 8.1150.2 Slots

Class Sparkle-Personality has no direct slots defined.

## **8.1151 Tootsville::Spawn-Terrain**

### **8.1151.1 Function**

Spawn-Terrain names an undocumented function, with lambda list (PLACE LATITUDE LONGITUDE).

### **8.1151.2 File**

Defined in file src/terrain.lisp.

## **8.1152 Tootsville::Split-Backtrace**

### **8.1152.1 Function**

Split-Backtrace names a function, with lambda list (STR):

Split a string backtrace into parts

### **8.1152.2 File**

Defined in file src/errors.lisp.

## **8.1153 Tootsville::Split-Plist**

### **8.1153.1 Function**

Split-Plist names a function, with lambda list (PLIST):

Split a PLIST into two lists, of keys and values.

### **8.1153.2 File**

Defined in file src/utls.lisp.

## 8.1154 Tootsville::Sploot-Personality

### 8.1154.1 Class

Sploot-Personality names a class, with one superclass: Section 8.1104 [TOOTSVILLE ROBOT-SPLOOT], page 1389.

This class defines a character named Sploot

### 8.1154.2 Slots

Class Sploot-Personality has no direct slots defined.

## **8.1155 Tootsville::Square**

### **8.1155.1 Function**

Square names an undocumented function, with lambda list (X).

### **8.1155.2 File**

Defined in file src/world.lisp.

## **8.1156 Tootsville::Ssl-Certificate**

### **8.1156.1 Function**

Ssl-Certificate names an undocumented function, with lambda list NIL.

### **8.1156.2 File**

Defined in file src/config.lisp.

## **8.1157 Tootsville::Ssl-Private-Key**

### **8.1157.1 Function**

Ssl-Private-Key names an undocumented function, with lambda list NIL.

### **8.1157.2 File**

Defined in file src/config.lisp.



## **8.1158 Tootsville::Stamp-Toot-Passport**

### **8.1158.1 Function**

Stamp-Toot-Passport names a function, with lambda list (TOOT STAMP):

Stamp the passport for TOOT with STAMP.

### **8.1158.2 File**

Defined in file src/passport.lisp.

## **8.1159 Tootsville::Standard-Log-File**

### **8.1159.1 Function**

Standard-Log-File names a function, with lambda list (LOG-DIR):

Get the pathname of the standard log file.

### **8.1159.2 File**

Defined in file src/logging.lisp.

## 8.1160 Tootsville::Start

### 8.1160.1 Function

Start names a function, with lambda list (&KEY (HOST 0.0.0.0) (PORT 5000) (FULLP T)):

Start a local Hunchentoot server on HOST and PORT.

HOST is an address of a live interface; PORT may be a port number.

The server will be started running on port PORT (default 5000) on HOST (default local-loopback-only address "localhost"). If an existing server is running, a restart will be presented to allow you to kill it (RESTART-SERVER).

When FULLP is true, a complete start-up including reading config files, connecting to the databases, power-on self-test, &c. will be performed.

In addition, if a TSL (SSL) certificate for this host appears to be present, created by Let's Encrypt, then a TLS acceptor will be started on a port as identified in the configuration file, if that port is available.

### 8.1160.2 File

Defined in file src/main.lisp.

## **8.1161 Tootsville::Start-Game-Metronome**

### **8.1161.1 Function**

Start-Game-Metronome names an undocumented function, with lambda list NIL.

### **8.1161.2 File**

Defined in file src/metronome.lisp.

## 8.1162 Tootsville::Start-Hunchentoot

### 8.1162.1 Function

Start-Hunchentoot names a function, with lambda list (&KEY (HOST localhost) (PORT 5000)):

Start a Hunchentoot server via Section 8.1160 [TOOTSVILLE START], page 1445, and fall through into a REPL to keep the process running.

### 8.1162.2 File

Defined in file src/main.lisp.

## 8.1163 Tootsville::Start-Production

### 8.1163.1 Function

Start-Production names a function, with lambda list (&KEY HOST PORT):

Start a Hunchentoot server via Section 8.1160 [TOOTSVILLE START], page 1445, and daemonize with Swank.

This is the entry point for running a Production, stand-alone server.

SBCL's Low-level Debugger is disabled, so crashes are instantly fatal, allowing SystemD to start a new instance in case of a fatal error.

### 8.1163.2 File

Defined in file src/main.lisp.

## **8.1164 Tootsville::Start-Purchase-Event**

### **8.1164.1 Function**

Start-Purchase-Event names a function, with lambda list (STORE-ITEM TOOT):

Start an event for TOOT to purchase STORE-ITEM

### **8.1164.2 File**

Defined in file src/quaestor.lisp.

## 8.1165 Tootsville::Start-Swank

### 8.1165.1 Function

Start-Swank names a function, with lambda list (&OPTIONAL (PORT (+ 46046 (\* 2 (RANDOM 500))))):

Starts a SWANK server on PORT.

Writes the port number to a file named after this (parent) process's PID.

### 8.1165.2 File

Defined in file src/main.lisp.



## 8.1166 Tootsville::Start-Tcp-Listener

### 8.1166.1 Function

Start-Tcp-Listener names a function, with lambda list (&OPTIONAL (HOST ::1) (PORT 2773)):

Start listening for TCP peers on interface HOST and PORT.

The default PORT is 2773.

### 8.1166.2 File

Defined in file src/tcp-stream.lisp.

## **8.1167 Tootsville::Stop**

### **8.1167.1 Function**

Stop names a function, with lambda list (&OPTIONAL (ACCEPTOR (FIRST \*ACCEPTORS\*))):

Stop the Hunchentoot server process started by Section 8.1160 [TOOTSVILLE START], page 1445,

### **8.1167.2 File**

Defined in file src/main.lisp.

## **8.1168 Tootsville::Stop-Game-Metronome**

### **8.1168.1 Function**

Stop-Game-Metronome names a function, with lambda list NIL:

Stop the metronome facility by canceling all tasks and stopping the metronome thread.

### **8.1168.2 File**

Defined in file src/metronome.lisp.

## **8.1169 Tootsville::Stop-Listening-For-Websockets**

### **8.1169.1 Function**

Stop-Listening-For-Websockets names a function, with lambda list NIL:

Stop listening for websocket connections and disable the maintenance thread.

### **8.1169.2 File**

Defined in file src/websockets.lisp.

## **8.1170 Tootsville::Stop-Production**

### **8.1170.1 Function**

Stop-Production names an undocumented function, with lambda list NIL.

### **8.1170.2 File**

Defined in file src/main.lisp.

## 8.1171 Tootsville::Store-Info

### 8.1171.1 Function

Store-Info names a function, with lambda list (STORE-ITEM):

Returns a structure describing STORE-ITEM.

This structure is a JSON-style Plist with the keys:

- id**           The unique store item ID. This is currently a UUID.
- template**   The Section 8.787 [TOOTSVILLE ITEM-TEMPLATE-INFO], page 1072, of this item
- qty**         The quantity (integer) of these items available in the store.
- price**       The price (in **currency** units) of the item
- currency**   The currency indicator. This will generally be one of
  - X-TVPN      Tootsville peanuts; or
  - X-FADU      Fairy dust

### 8.1171.2 File

Defined in file src/items.lisp.

## 8.1172 Tootsville::Store-Item

### 8.1172.1 Class

Store-Item names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.1172.2 Slots

Class Store-Item has 5 direct slot definitions:

Uuid

Template

Qty

Price

Currency

## **8.1173 Tootsville::Store-Item-Currency**

### **8.1173.1 Function**

Store-Item-Currency names an undocumented function, with lambda list (INSTANCE).

### **8.1173.2 File**

Defined in file src/db/friendly.lisp.

### **8.1173.3 SetF Function**

(SETF Store-Item-Currency) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1173.4 File**

Defined in file src/db/friendly.lisp.



## **8.1174 Tootsville::Store-Item-P**

### **8.1174.1 Function**

Store-Item-P names an undocumented function, with lambda list (OBJECT).

### **8.1174.2 File**

Defined in file src/db/friendly.lisp.

## **8.1175 Tootsville::Store-Item-Price**

### **8.1175.1 Function**

Store-Item-Price names an undocumented function, with lambda list (INSTANCE).

### **8.1175.2 File**

Defined in file src/db/friendly.lisp.

### **8.1175.3 SetF Function**

(SETF Store-Item-Price) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1175.4 File**

Defined in file src/db/friendly.lisp.

## **8.1176 Tootsville::Store-Item-Qty**

### **8.1176.1 Function**

Store-Item-Qty names an undocumented function, with lambda list (INSTANCE).

### **8.1176.2 File**

Defined in file src/db/friendly.lisp.

### **8.1176.3 SetF Function**

(SETF Store-Item-Qty) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1176.4 File**

Defined in file src/db/friendly.lisp.

## **8.1177 Tootsville::Store-Item-Template**

### **8.1177.1 Function**

Store-Item-Template names an undocumented function, with lambda list (INSTANCE).

### **8.1177.2 File**

Defined in file src/db/friendly.lisp.

### **8.1177.3 SetF Function**

(SETF Store-Item-Template) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1177.4 File**

Defined in file src/db/friendly.lisp.

## 8.1178 Tootsville::Store-Item-Uuid

### 8.1178.1 Function

Store-Item-Uuid names an undocumented function, with lambda list (INSTANCE).

### 8.1178.2 File

Defined in file src/db/friendly.lisp.

### 8.1178.3 SetF Function

(SETF Store-Item-Uuid) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1178.4 File

Defined in file src/db/friendly.lisp.

## **8.1179 Tootsville::String-All-Alpha-Chars-P**

### **8.1179.1 Function**

String-All-Alpha-Chars-P names a function, with lambda list (S):

Is S a string of only alphabetical characters?

### **8.1179.2 File**

Defined in file `src/types/string-characteristics.lisp`.

## **8.1180 Tootsville::String-Length-2-P**

### **8.1180.1 Function**

String-Length-2-P names a function, with lambda list (S):

Is S a string of length 2?

### **8.1180.2 File**

Defined in file `src/types/string-characteristics.lisp`.

## **8.1181 Tootsville::Strip-After-Sem**

### **8.1181.1 Function**

Strip-After-Sem names an undocumented function, with lambda list (S).

### **8.1181.2 File**

Defined in file src/acceptor.lisp.



## **8.1182 Tootsville::Subheader-Field**

### **8.1182.1 Function**

Subheader-Field names an undocumented function, with lambda list (HEADER-ASSOC LABEL).

### **8.1182.2 File**

Defined in file src/auth/auth-firebase.lisp.

## 8.1183 Tootsville::Sun-Position

### 8.1183.1 Function

Sun-Position names a function, with lambda list (&OPTIONAL (TIME (GET-UNIVERSAL-TIME))):

The position (X,Y) of the sun as a list of 2 elements

### 8.1183.2 File

Defined in file src/weather/sun-moon.lisp.

## **8.1184 Tootsville::Superstar-Personality**

### **8.1184.1 Class**

Superstar-Personality names a class, with one superclass: Section 8.1105 [TOOTSVILLE ROBOT-SUPERSTAR], page 1390.

This class defines a character named Superstar

### **8.1184.2 Slots**

Class Superstar-Personality has no direct slots defined.

## **8.1185 Tootsville::Swank-Connected-P**

### **8.1185.1 Function**

Swank-Connected-P names a function, with lambda list NIL:

Is Swank currently connected to this Lisp image?

### **8.1185.2 File**

Defined in file src/main.lisp.

## **8.1186 Tootsville::Sync**

### **8.1186.1 Function**

Sync names an undocumented function, with lambda list NIL.

### **8.1186.2 File**

Defined in file src/utls.lisp.

## **8.1187 Tootsville::Take-Item**

### **8.1187.1 Function**

Take-Item names a function, with lambda list (ITEM RECIPIENT):

RECIPIENT becomes the new owner of ITEM.

The RECIPIENT Toot must be close enough to pick up ITEM, and ITEM must be in the world, and not owned by any other player.

### **8.1187.2 File**

Defined in file src/items.lisp.

## **8.1188 Tootsville::Tcp-Bandwidth-Record**

### **8.1188.1 Function**

Tcp-Bandwidth-Record names an undocumented function, with lambda list (MESSAGE &OPTIONAL (MULTIPLIER 1)).

### **8.1188.2 File**

Defined in file src/tcp-stream.lisp.

## **8.1189 Tootsville::Tcp-Broadcast**

### **8.1189.1 Function**

Tcp-Broadcast names an undocumented function, with lambda list (MESSAGE).

### **8.1189.2 File**

Defined in file src/tcp-stream.lisp.



## 8.1190 Tootsville::Tcp-Client

### 8.1190.1 Class

Tcp-Client names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.1190.2 Slots

Class Tcp-Client has 4 direct slot definitions:

Socket

Buffer

Expected-Length

Peer

## **8.1191 Tootsville::Tcp-Client-Buffer**

### **8.1191.1 Function**

Tcp-Client-Buffer names an undocumented function, with lambda list (INSTANCE).

### **8.1191.2 File**

Defined in file src/tcp-stream.lisp.

### **8.1191.3 SetF Function**

(SETF Tcp-Client-Buffer) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1191.4 File**

Defined in file src/tcp-stream.lisp.

## **8.1192 Tootsville::Tcp-Client-Expected-Length**

### **8.1192.1 Function**

Tcp-Client-Expected-Length names an undocumented function, with lambda list (INSTANCE).

### **8.1192.2 File**

Defined in file src/tcp-stream.lisp.

### **8.1192.3 SetF Function**

(SETF Tcp-Client-Expected-Length) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1192.4 File**

Defined in file src/tcp-stream.lisp.

## **8.1193 Tootsville::Tcp-Client-P**

### **8.1193.1 Function**

Tcp-Client-P names an undocumented function, with lambda list (OBJECT).

### **8.1193.2 File**

Defined in file src/tcp-stream.lisp.

## **8.1194 Tootsville::Tcp-Client-Peer**

### **8.1194.1 Function**

Tcp-Client-Peer names an undocumented function, with lambda list (INSTANCE).

### **8.1194.2 File**

Defined in file src/tcp-stream.lisp.

### **8.1194.3 SetF Function**

(SETF Tcp-Client-Peer) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1194.4 File**

Defined in file src/tcp-stream.lisp.

## **8.1195 Tootsville::Tcp-Client-Socket**

### **8.1195.1 Function**

Tcp-Client-Socket names an undocumented function, with lambda list (INSTANCE).

### **8.1195.2 File**

Defined in file src/tcp-stream.lisp.

### **8.1195.3 SetF Function**

(SETF Tcp-Client-Socket) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1195.4 File**

Defined in file src/tcp-stream.lisp.

## **8.1196 Tootsville::Tcp-Format-Error**

### **8.1196.1 Function**

Tcp-Format-Error names a function, with lambda list (TCP-CLIENT):

Send a format error to TCP-CLIENT.

This is the character EM, End of Medium, ASCII value 25 (decimal).

### **8.1196.2 File**

Defined in file src/tcp-stream.lisp.

## **8.1197 Tootsville::Tcp-Handle-Peer-Request**

### **8.1197.1 Function**

Tcp-Handle-Peer-Request names an undocumented function, with lambda list (MESSAGE PEER).

### **8.1197.2 File**

Defined in file src/tcp-stream.lisp.



## **8.1198 Tootsville::Tcp-Process-Packet**

### **8.1198.1 Function**

Tcp-Process-Packet names an undocumented function, with lambda list (PACKET TCP-CLIENT).

### **8.1198.2 File**

Defined in file src/tcp-stream.lisp.

## **8.1199 Tootsville::Tcp-Reply**

### **8.1199.1 Function**

Tcp-Reply names an undocumented function, with lambda list (MESSAGE TCP-CLIENT).

### **8.1199.2 File**

Defined in file src/tcp-stream.lisp.

## **8.1200 Tootsville::Tcp-Socket-Input**

### **8.1200.1 Function**

Tcp-Socket-Input names an undocumented function, with lambda list (TCP-CLIENT).

### **8.1200.2 File**

Defined in file src/tcp-stream.lisp.

## 8.1201 Tootsville::Tcp-Stream-Authenticate

### 8.1201.1 Function

Tcp-Stream-Authenticate names a function, with lambda list (CLIENT AUTH\$):

Private server-to-server messaging authentication.

Tunnelled over SSH, so a simple non-cryptographically-secure authentication is all that's performed here.

TODO: This is not implemented.

### 8.1201.2 File

Defined in file src/tcp-stream.lisp.

## 8.1202 Tootsville::Tcp-Unicast

### 8.1202.1 Function

Tcp-Unicast names a function, with lambda list (MESSAGE TCP-CLIENT):

Writes MESSAGE to TCP-CLIENT.

MESSAGE is encoded with a SOH (start of heading, ASCII value 1), followed by the length of the message in base-36, then STX (start of text, ASCII value 2), the message itself, and a final ETX (end of text, ASCII value 3).

### 8.1202.2 File

Defined in file src/tcp-stream.lisp.

## **8.1203 Tootsville::Template->Openapi**

### **8.1203.1 Function**

Template->Openapi names a function, with lambda list (TEMPLATE):

Convert URI TEMPLATE into an OpenAPI template string.

### **8.1203.2 File**

Defined in file src/endpoints/slash-meta-game.lisp.

## 8.1204 Tootsville::Template-Match

### 8.1204.1 Function

Template-Match names a function, with lambda list (TEMPLATE LIST):

Attempt to match a template list against a split-down URI.

The template list consists of strings, which must match exactly, or symbols, in which case any string will match. The values to which symbols are bound are returned sequentially, like positional parameters.

### 8.1204.2 File

Defined in file src/acceptor.lisp.

## 8.1205 Tootsville::Terrain

### 8.1205.1 Function

Terrain names a function, with lambda list (PLACE LATITUDE LONGITUDE):

Obtain the terrain tile in PLACE at LATITUDE, LONGITUDE

PLACE is one of :chor, :Moon, :othm, :pink, :orbit.

LATITUDE and LONGITUDE must be aligned to 200m increments.

### 8.1205.2 File

Defined in file src/terrain.lisp.



## 8.1206 Tootsville::Terrain-Db-Key

### 8.1206.1 Function

Terrain-Db-Key names an undocumented function, with lambda list (PLACE LATITUDE LONGITUDE).

### 8.1206.2 File

Defined in file src/terrain.lisp.

## **8.1207 Tootsville::Terrain-Exists-P**

### **8.1207.1 Function**

Terrain-Exists-P names a function, with lambda list (PLACE LATITUDE LONGITUDE):

If terrain has been previously defined at the tile given, return it.

Use Section 8.1205 [TOOTSVILLE TERRAIN], page 1490, generally instead.

### **8.1207.2 File**

Defined in file src/terrain.lisp.

## 8.1208 Tootsville::Terrain-Height

### 8.1208.1 Class

Terrain-Height names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.1208.2 Slots

Class Terrain-Height has 4 direct slot definitions:

World

Latitude

Longitude

Terrain

## **8.1209 Tootsville::Terrain-Height-Latitude**

### **8.1209.1 Function**

Terrain-Height-Latitude names an undocumented function, with lambda list (INSTANCE).

### **8.1209.2 File**

Defined in file src/db/friendly.lisp.

### **8.1209.3 SetF Function**

(SETF Terrain-Height-Latitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1209.4 File**

Defined in file src/db/friendly.lisp.

## 8.1210 Tootsville::Terrain-Height-Longitude

### 8.1210.1 Function

Terrain-Height-Longitude names an undocumented function, with lambda list (INSTANCE).

### 8.1210.2 File

Defined in file src/db/friendly.lisp.

### 8.1210.3 SetF Function

(SETF Terrain-Height-Longitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1210.4 File

Defined in file src/db/friendly.lisp.

## **8.1211 Tootsville::Terrain-Height-P**

### **8.1211.1 Function**

Terrain-Height-P names an undocumented function, with lambda list (OBJECT).

### **8.1211.2 File**

Defined in file src/db/friendly.lisp.

## **8.1212 Tootsville::Terrain-Height-Terrain**

### **8.1212.1 Function**

Terrain-Height-Terrain names an undocumented function, with lambda list (INSTANCE).

### **8.1212.2 File**

Defined in file src/db/friendly.lisp.

### **8.1212.3 SetF Function**

(SETF Terrain-Height-Terrain) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1212.4 File**

Defined in file src/db/friendly.lisp.

## **8.1213 Tootsville::Terrain-Height-World**

### **8.1213.1 Function**

Terrain-Height-World names an undocumented function, with lambda list (INSTANCE).

### **8.1213.2 File**

Defined in file src/db/friendly.lisp.

### **8.1213.3 SetF Function**

(SETF Terrain-Height-World) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1213.4 File**

Defined in file src/db/friendly.lisp.



## **8.1214 Tootsville::Terrain/ Add-Cactus**

### **8.1214.1 Function**

Terrain/ Add-Cactus names a function, with lambda list NIL:

Add a cactus

### **8.1214.2 File**

Defined in file src/terrain.lisp.

## **8.1215 Tootsville::Terrain/ Add-Flowers**

### **8.1215.1 Function**

Terrain/ Add-Flowers names a function, with lambda list NIL:

Add a random cluster of appropriate flowers or herbs.

### **8.1215.2 File**

Defined in file src/terrain.lisp.

## **8.1216 Tootsville::Terrain/ Add-Log**

### **8.1216.1 Function**

Terrain/ Add-Log names a function, with lambda list NIL:

    Adds a fallen log or similar feature.

### **8.1216.2 File**

Defined in file src/terrain.lisp.

## **8.1217 Tootsville::Terrain/ Add-Mushrooms**

### **8.1217.1 Function**

Terrain/ Add-Mushrooms names a function, with lambda list NIL:

Add a cluster of mushrooms or similar.

### **8.1217.2 File**

Defined in file src/terrain.lisp.

## **8.1218 Tootsville::Terrain/ Add-Shaddow-Bush**

### **8.1218.1 Function**

Terrain/ Add-Shaddow-Bush names a function, with lambda list NIL:

Add a Shaddow bush to the area

### **8.1218.2 File**

Defined in file src/terrain.lisp.

## **8.1219 Tootsville::Terrain/ Add-Shaddow-Pit**

### **8.1219.1 Function**

Terrain/ Add-Shaddow-Pit names a function, with lambda list NIL:

Add a Shaddow pit to the area

### **8.1219.2 File**

Defined in file src/terrain.lisp.

## **8.1220 Tootsville::Terrain/ Add-Shaddow-Stalagmite**

### **8.1220.1 Function**

Terrain/ Add-Shaddow-Stalagmite names a function, with lambda list NIL:

Add a Shaddow stalagmite to the area

### **8.1220.2 File**

Defined in file src/terrain.lisp.

## **8.1221 Tootsville::Terrain/ Add-Small-Pond**

### **8.1221.1 Function**

Terrain/ Add-Small-Pond names a function, with lambda list NIL:

    Create a pool of water smaller than the tile and contained within it. TODO

### **8.1221.2 File**

Defined in file src/terrain.lisp.



## **8.1222 Tootsville::Terrain/ Add-Tree**

### **8.1222.1 Function**

Terrain/ Add-Tree names a function, with lambda list NIL:

    Add a random tree or bush.

### **8.1222.2 File**

Defined in file src/terrain.lisp.

## **8.1223 Tootsville::Terrain/ Connect-Streams**

### **8.1223.1 Function**

Terrain/Connect-Streams names an undocumented function, with lambda list NIL.

### **8.1223.2 File**

Defined in file src/terrain.lisp.

## 8.1224 Tootsville::Terrain/ Stream-Present-P

### 8.1224.1 Function

Terrain/ Stream-Present-P names a function, with lambda list NIL:

Does a stream bisect the currently-active space?

Should return true if a body of water exists which enters the space from any side and bisects the space into two disjoint land areas. Terminus of a stream or completely underwater are not “streams” by this definition.

### 8.1224.2 File

Defined in file src/terrain.lisp.

## **8.1225 Tootsville::Test**

### **8.1225.1 Variable**

Test names an undocumented variable with the value NIL

## 8.1226 Tootsville::Texi-Ref

### 8.1226.1 Function

Texi-Ref names a function, with lambda list (STRING):

Given STRING is a TeXInfo text, replace any ‘single-quoted’ links.

A single-quoted reference to a Lisp symbol will be replaced with a hyperlink to that section of the manual, or given an annotation if it is not in this manual.

References to “Tootsville.” are assumed to be valid Javascripts.

### 8.1226.2 File

Defined in file src/write-docs-2.lisp.

## **8.1227 Tootsville::Three-Chars-In-A-Row-P**

### **8.1227.1 Function**

Three-Chars-In-A-Row-P names a function, with lambda list (STRING &OPTIONAL CHAR-BAG):

Do any three characters in CHAR-BAG occur together in STRING?

If CHAR-BAG is NIL, then any character that occurs three times matching itself returns true.

### **8.1227.2 File**

Defined in file src/types/string-characteristics.lisp.

## 8.1228 Tootsville::Tick-Weather-Day

### 8.1228.1 Function

Tick-Weather-Day names a function, with lambda list NIL:

- Precipitation chances are highest in the third months – Inunguis, Senecalensis, Elephas, and Tethytheria – peaking at the 15th of each third month. Thus, the least chance of precipitation is around the 1st of the second month of each quarter – 1 Dugon, Hyrodamalis, Luxodonta, and Dendrohyrax
- Winter precipitation (from about 1 Tethytheria to about 30 Dugon) will tend to be coming from the mountains to the sea, and bring snow, with a corresponding drop in temperature. Summer precipitation (from about 1 Senecalensis to 30 Luxodonta) will tend to come from the south seas, and bring warmer temperatures. The temperature won't be generally affected at all by precipitation during the spring and autumn months.

### 8.1228.2 File

Defined in file `src/weather/weather.lisp`.

## **8.1229 Tootsville::Tick-Weather-Minute**

### **8.1229.1 Function**

Tick-Weather-Minute names an undocumented function, with lambda list NIL.

### **8.1229.2 File**

Defined in file src/weather/weather.lisp.



## 8.1230 Tootsville::Toot

### 8.1230.1 Function

Toot names a function, with lambda list (IDENTIFIER):

Find the Toot associated with IDENTIFIER.

### 8.1230.2 File

Defined in file src/websockets.lisp.

### 8.1230.3 SetF Function

(SETF Toot) names an undocumented function, with lambda list (NEW-VALUE OBJECT).

### 8.1230.4 Class

Toot names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.1230.5 Slots

Class Toot has 14 direct slot definitions:

Uuid

Name

Pattern

Base-Color

Pattern-Color

Pad-Color

Avatar

Player

Child-Code

Last-Active

Note

Avatar-Scale-X

Avatar-Scale-Y

Avatar-Scale-Z

## **8.1231 Tootsville::Toot-Avatar**

### **8.1231.1 Function**

Toot-Avatar names an undocumented function, with lambda list (INSTANCE).

### **8.1231.2 File**

Defined in file src/db/friendly.lisp.

### **8.1231.3 SetF Function**

(SETF Toot-Avatar) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1231.4 File**

Defined in file src/db/friendly.lisp.

## **8.1232 Tootsville::Toot-Avatar-Scale-X**

### **8.1232.1 Function**

Toot-Avatar-Scale-X names an undocumented function, with lambda list (INSTANCE).

### **8.1232.2 File**

Defined in file src/db/friendly.lisp.

### **8.1232.3 SetF Function**

(SETF Toot-Avatar-Scale-X) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1232.4 File**

Defined in file src/db/friendly.lisp.

## **8.1233 Tootsville::Toot-Avatar-Scale-Y**

### **8.1233.1 Function**

Toot-Avatar-Scale-Y names an undocumented function, with lambda list (INSTANCE).

### **8.1233.2 File**

Defined in file src/db/friendly.lisp.

### **8.1233.3 SetF Function**

(SETF Toot-Avatar-Scale-Y) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1233.4 File**

Defined in file src/db/friendly.lisp.

## **8.1234 Tootsville::Toot-Avatar-Scale-Z**

### **8.1234.1 Function**

Toot-Avatar-Scale-Z names an undocumented function, with lambda list (INSTANCE).

### **8.1234.2 File**

Defined in file src/db/friendly.lisp.

### **8.1234.3 SetF Function**

(SETF Toot-Avatar-Scale-Z) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1234.4 File**

Defined in file src/db/friendly.lisp.

## **8.1235 Tootsville::Toot-Base-Color**

### **8.1235.1 Function**

Toot-Base-Color names an undocumented function, with lambda list (INSTANCE).

### **8.1235.2 File**

Defined in file src/db/friendly.lisp.

### **8.1235.3 SetF Function**

(SETF Toot-Base-Color) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1235.4 File**

Defined in file src/db/friendly.lisp.

## **8.1236 Tootsville::Toot-Base-Color-Name**

### **8.1236.1 Type**

Toot-Base-Color-Name names a TYPE:

A string designator which describes a valid color for a Toot's base color.

See '+TOOT-BASE-COLOR-NAMES' for the list.

## 8.1237 Tootsville::Toot-Base-Color-Name-P

### 8.1237.1 Function

Toot-Base-Color-Name-P names an undocumented function, with lambda list (&REST ARGUMENTS).

### 8.1237.2 File

Defined in file `quicklisp/dists/quicklisp/software/fare-memoization-20180430-git/memoization.lisp`.



## 8.1238 Tootsville::Toot-Buddy-List

### 8.1238.1 Function

Toot-Buddy-List names an undocumented function, with lambda list (&OPTIONAL (TOOT \*TOOT\*)).

### 8.1238.2 File

Defined in file src/infinity/legacy-commands.lisp.

## **8.1239 Tootsville::Toot-Can-Afford-P**

### **8.1239.1 Function**

Toot-Can-Afford-P names a function, with lambda list (TOOT STORE-ITEM):

Whether TOOT can afford STORE-ITEM

### **8.1239.2 File**

Defined in file src/quaestor.lisp.

## 8.1240 Tootsville::Toot-Chat-Background-Color

### 8.1240.1 Function

Toot-Chat-Background-Color names a function, with lambda list (TOOT):

The background color of a Toot's speech balloon in normal speech.

Shouting and whispering should alter this color appropriately.

Obtained via Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534.

Always white at present (2.0). This should not be trusted to be a constant; it should be updated in a later release.

### 8.1240.2 File

Defined in file src/toots.lisp.

## **8.1241 Tootsville::Toot-Chat-Foreground-Color**

### **8.1241.1 Function**

Toot-Chat-Foreground-Color names a function, with lambda list (TOOT):

The foreground (text) color of a Toot's speech balloon in normal speech.

Shouting and whispering should alter this color appropriately.

Obtained via Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534.

Always black at present (2.0). This should not be trusted to be a constant; it should be updated in a later release.

### **8.1241.2 File**

Defined in file src/toots.lisp.

## **8.1242 Tootsville::Toot-Child-Code**

### **8.1242.1 Function**

Toot-Child-Code names an undocumented function, with lambda list (INSTANCE).

### **8.1242.2 File**

Defined in file src/db/friendly.lisp.

### **8.1242.3 SetF Function**

(SETF Toot-Child-Code) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1242.4 File**

Defined in file src/db/friendly.lisp.

## **8.1243 Tootsville::Toot-Childp**

### **8.1243.1 Function**

Toot-Childp names an undocumented function, with lambda list (TOOT).

### **8.1243.2 File**

Defined in file src/toots.lisp.

## **8.1244 Tootsville::Toot-Clothes+Pattern**

### **8.1244.1 Function**

Toot-Clothes+Pattern names a function, with lambda list (TOOT):

The clothes (including Pivitz) and pattern that TOOT is wearing.

### **8.1244.2 File**

Defined in file src/toots.lisp.

## **8.1245 Tootsville::Toot-Contacts**

### **8.1245.1 Function**

Toot-Contacts names an undocumented function, with lambda list (TOOT).

### **8.1245.2 File**

Defined in file `src/contacts.lisp`.



## **8.1246 Tootsville::Toot-Equipped-Item**

### **8.1246.1 Function**

Toot-Equipped-Item names an undocumented function, with lambda list (TOOT).

### **8.1246.2 File**

Defined in file src/toots.lisp.

## **8.1247 Tootsville::Toot-Fairy-Dust**

### **8.1247.1 Function**

Toot-Fairy-Dust names a function, with lambda list (TOOT):

Compute the total balance of fairy dust that TOOT has earned over the course of the game.

### **8.1247.2 File**

Defined in file src/quaestor.lisp.

## 8.1248 Tootsville::Toot-Has-Item-P

### 8.1248.1 Function

Toot-Has-Item-P names a function, with lambda list (ITEM-TEMPLATE-ID &OPTIONAL (TOOT \*TOOT\*)):

A generalize boolean indicating whether TOOT has any item based upon ITEM-TEMPLATE-ID

Calls Section 8.1250 [TOOTSVILLE TOOT-INVENTORY], page 1537, to benefit from caching.

### 8.1248.2 File

Defined in file src/items.lisp.

## 8.1249 Tootsville::Toot-Info

### 8.1249.1 Function

Toot-Info names a function, with lambda list (TOOT &OPTIONAL (PRIVATEP (AND \*USER\* (UUID= (PERSON-UUID \*USER\*) (TOOT-PLAYER TOOT))))):

Returns a JSON-compatible structure which describes TOOT.

If PRIVATEP, then private information (normally only visible to that Toot's user) is returned; otherwise, private information is dummied out or absent.

This data is returned by various functions, including Section 8.691 [TOOTSVILLE INFINITY-FINGER], page 953, or Section 8.742 [TOOTSVILLE INFINITY-WARDROBE], page 1026.

### 8.1249.2 Data Structure

**name** The name of the Toot character. See Section 8.1254 [TOOTSVILLE TOOT-NAME], page 1541,

**userName** Also the name of the Toot character, in the form in which it should appear on an avatar label. Notably, this means that child or sensitive users will have a black diamond prefixed to the name. See Section 8.1272 [TOOTSVILLE TOOT-PRESENTATION-NAME], page 1559,

**avatar** The base filename of the avatar. The actual URL for the avatar model will always be <https://jumbo.tootsville.org/Assets/Avatars/5/avatar.babylon>. See Section 8.1231 [TOOTSVILLE TOOT-AVATAR], page 1516, to obtain the avatar's ID, and Section 8.141 [TOOTSVILLE AVATAR-MONIKER], page 395, to obtain the name from that ID.

**chatFG** The foreground (text) color of this character's chat messages. See Section 8.1241 [TOOTSVILLE TOOT-CHAT-FOREGROUND-COLOR], page 1526.

**chatBG** The background color of the speech balloons behind the character's chat messages. See Section 8.1240 [TOOTSVILLE TOOT-CHAT-BACKGROUND-COLOR], page 1525.

#### avatarClass

This is a legacy object which describes the avatar in play, in theory. It has the following attributes: **id**, the unique ID for the avatar; **title**, always the same as **avatar** URL base name; **filename**, also the same; **forFree**, always **true**; **forPaid**, always **false**.

#### avatarClass\_B, baseColor

The base color for the avatar's skin. Sent as two identical values. See Section 8.1235 [TOOTSVILLE TOOT-BASE-COLOR], page 1520,

#### avatarClass\_P, patternColor

The color for the avatars's pattern, if any. Sent as two identical values. See Section 8.1263 [TOOTSVILLE TOOT-PATTERN-COLOR], page 1550,

#### avatarClass\_E, padColor

The color for the avatar's pad or "extra" color. Sent as two identical values. See Section 8.1257 [TOOTSVILLE TOOT-PAD-COLOR], page 1544,

<code>format</code>	Always the same as <code>avatar</code> now.
<code>colors</code>	The list of base, pattern, and extra color, a third time, as an array-like object; keys are 0 for base color, 1 for pad color, and 2 for pattern color.
<code>inRoom</code>	No longer returned; always reads exactly “@Tootsville”
<code>vars</code>	No longer returned; always nil.
<code>clothes</code>	The clothing currently being worn by the character. For legacy reasons, the character’s pattern is repeated here. Pivitz are considered clothes. See Section 8.1244 [TOOTSVILLE TOOT-CLOTHES+PATTERN], page 1529,
<code>pattern</code>	The name of the pattern of the avatar, if any. See Section 8.1262 [TOOTSVILLE TOOT-PATTERN], page 1549,
<code>gameItem</code>	The item currently held in the character’s TRUNK or HAND slot, as appropriate to the avatar model, if any. See Section 8.1246 [TOOTSVILLE TOOT-EQUIPPED-ITEM], page 1531,
<code>uuid, id</code>	The Toot character’s UUID. See Section 8.1290 [TOOTSVILLE TOOT-UUID], page 1577,
<code>equip</code>	If this is the requestor’s Toot, a set of all inventory (equipment) as per ‘TOOT-ITEM-INFO’
<code>childP</code>	True if the Toot represents a child player. See Section 8.1243 [TOOTSVILLE TOOT-CHILDP], page 1528,
<code>childRequest</code>	If there is an active or pending request to play from this child, this object will be attached. It contains <code>uuid</code> , the time the request was <code>placedAt</code> , when it was <code>allowedAt</code> or <code>deniedAt</code> , how long it was <code>allowedFor</code> , and any <code>response</code> text.
<code>childCode</code>	Only available to the user owning the Toot, this is the code to log in as the child Toot. See Section 8.1242 [TOOTSVILLE TOOT-CHILD-CODE], page 1527,
<code>sensitiveP</code>	True if the Toot represents a sensitive user or a child. See Section 8.978 [TOOTSVILLE PERSON-SENSITIVEP], page 1263,
<code>scaling</code>	Scaling of the avatar in each of x, y, and z dimensions. See Section 8.1232 [TOOTSVILLE TOOT-AVATAR-SCALE-X], page 1517, Section 8.1233 [TOOTSVILLE TOOT-AVATAR-SCALE-Y], page 1518, and Section 8.1234 [TOOTSVILLE TOOT-AVATAR-SCALE-Z], page 1519,

### 8.1249.3 Changes from 1.0 to 1.1

The `avatarClass` object used to have fields `s`, which is the same as `title`; `forVIT`, which is the same as `forPaid`; and `avatarClassID`, which is the same as `id`. The renamed fields were supported under both names in 1.1 or 1.2 based on the setting of the global configuration variable `org.starhope.appius.events.format1.0`.

### 8.1249.4 Changes from 1.1 to 1.2

Added `scaling` for “Magic Toots.”

### 8.1249.5 Changes from 1.2 to 2.0

- Added `name`
- Dropped backwards compatibility with the `avatarClass` object from 1.0
- `id` now returns a UUID, not a fixnum integer.
- Avatars are now Babylon 3D models, not Flash objects, and are retrieved from a different URL pattern.
- Prepend black diamonds to `userName` for children or sensitive users.
- Always returns white and black for `chatFG` and `chatBG`, as “Magic Toot” colors are not currently supported in 2.0; they may return in 2.1 or later.
- Added `baseColor`, `patternColor`, and `padColor` names in parallel to existing, now deprecated, `colors` values.
- The `avatarClass_B,P,E` values, which used to reflect default colors for an avatar model, are now just the Toot’s current colors.
- Added `uuid`, `childP`, `childRequest`, `sensitiveP`, and `lastSeen`
- When the requestor owns this Toot, added `note`, `childCode`, `peanuts`, `fairyDust`,
- `inRoom` always returns “@Tootsville”.
- `vars` always returns nil.

See also Deprecation section below.

### 8.1249.6 Deprecation

The following elements are deprecated and will be removed in a future revision:

`id`            use `uuid` in future.

`avatarClass_B,_P,_E` and `colors`

                Deprecated in favor of `baseColor`, `patternColor`, `padColor`.

`avatarClass`

                This is deprecated and will be removed in future. Its purpose is better served by other fields already in the structure.

`format`        This is deprecated in favor of `avatar`

### 8.1249.7 Obtaining Toot Information

Avatar information is available through several channels.

`/toots/Toot-Name`

                Fetch only the avatar information for a single Toot from this endpoint

Section 8.691 [TOOTSVILLE INFINITY-FINGER], page 953

                Fetch avatar information for a list of Toots.

### 8.1249.8 File

Defined in file `src/toots.lisp`.

## 8.1250 Tootsville::Toot-Inventory

### 8.1250.1 Function

Toot-Inventory names a function, with lambda list (&OPTIONAL (TOOT \*TOOT\*) &KEY PRIVATEP):

The inventory of TOOT, possibly including PRIVATEP items.

When PRIVATEP is false (default), only the inventory items which are equipped will be enumerated.

Returns a list of ITEM objects.

### 8.1250.2 File

Defined in file src/items.lisp.

## 8.1251 Tootsville::Toot-Join-Message

### 8.1251.1 Function

Toot-Join-Message names a function, with lambda list (&OPTIONAL (TOOT \*TOOT\*) (WORLD CHOR)):

Send joinOK message for TOOT

### 8.1251.2 File

Defined in file src/websockets.lisp.



## 8.1252 Tootsville::Toot-Last-Active

### 8.1252.1 Function

Toot-Last-Active names an undocumented function, with lambda list (INSTANCE).

### 8.1252.2 File

Defined in file src/db/friendly.lisp.

### 8.1252.3 SetF Function

(SETF Toot-Last-Active) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1252.4 File

Defined in file src/db/friendly.lisp.

## 8.1253 Tootsville::Toot-List-Message

### 8.1253.1 Function

Toot-List-Message names a function, with lambda list NIL:

Send a player (user) their list of Toots.

Used primarily in the login process. Might also be used for gifting inventory back-and-forth later.

### 8.1253.2 Format

```
{ from: "tootList",  
  status: true,  
  toots: [ TOOT-INFO, ... ] }
```

The value of `toots` is an array (list) of Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534, ordered by the time that the Toot was last active in the game, most recent to least recent. Clients are encouraged to display the list of Toots in this order.

If the player has no Toots yet, returns a 404 with `status: false`.

### 8.1253.3 File

Defined in file `src/infinity/new-commands-20.lisp`.

## 8.1254 Tootsville::Toot-Name

### 8.1254.1 Function

Toot-Name names an undocumented function, with lambda list (INSTANCE).

### 8.1254.2 File

Defined in file src/db/friendly.lisp.

### 8.1254.3 SetF Function

(SETF Toot-Name) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1254.4 File

Defined in file src/db/friendly.lisp.

### 8.1254.5 Type

Toot-Name names a TYPE:

A name that can be used for a Toot character.

See Section 8.1008 [TOOTSVILLE POTENTIAL-TOOT-NAME-P], page 1293.

## **8.1255 Tootsville::Toot-Note**

### **8.1255.1 Function**

Toot-Note names an undocumented function, with lambda list (INSTANCE).

### **8.1255.2 File**

Defined in file src/db/friendly.lisp.

### **8.1255.3 SetF Function**

(SETF Toot-Note) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1255.4 File**

Defined in file src/db/friendly.lisp.

## **8.1256 Tootsville::Toot-P**

### **8.1256.1 Function**

Toot-P names an undocumented function, with lambda list (OBJECT).

### **8.1256.2 File**

Defined in file src/db/friendly.lisp.

## **8.1257 Tootsville::Toot-Pad-Color**

### **8.1257.1 Function**

Toot-Pad-Color names an undocumented function, with lambda list (INSTANCE).

### **8.1257.2 File**

Defined in file src/db/friendly.lisp.

### **8.1257.3 SetF Function**

(SETF Toot-Pad-Color) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1257.4 File**

Defined in file src/db/friendly.lisp.

## **8.1258 Tootsville::Toot-Pad-Color-Name**

### **8.1258.1 Type**

Toot-Pad-Color-Name names a TYPE:

A color name that can be used for Toot foot pads and nose tip.

Formerly known as the “extra color” of the avatar.

## 8.1259 Tootsville::Toot-Pad-Color-Name-P

### 8.1259.1 Function

Toot-Pad-Color-Name-P names an undocumented function, with lambda list (&REST ARGUMENTS).

### 8.1259.2 File

Defined in file `quicklisp/dists/quicklisp/software/fare-memoization-20180430-git/memoization.lisp`.



## **8.1260 Tootsville::Toot-Passport-Stamped-P**

### **8.1260.1 Function**

Toot-Passport-Stamped-P names a function, with lambda list (TOOT STAMP):

Has TOOT's passport been stamped with STAMP?

### **8.1260.2 File**

Defined in file src/passport.lisp.

## **8.1261 Tootsville::Toot-Passport-Stamps**

### **8.1261.1 Function**

Toot-Passport-Stamps names a function, with lambda list (TOOT):

Enumerate the stamp names on TOOT's passport

### **8.1261.2 File**

Defined in file src/passport.lisp.

## **8.1262 Tootsville::Toot-Pattern**

### **8.1262.1 Function**

Toot-Pattern names an undocumented function, with lambda list (INSTANCE).

### **8.1262.2 File**

Defined in file src/db/friendly.lisp.

### **8.1262.3 SetF Function**

(SETF Toot-Pattern) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1262.4 File**

Defined in file src/db/friendly.lisp.

## **8.1263 Tootsville::Toot-Pattern-Color**

### **8.1263.1 Function**

Toot-Pattern-Color names an undocumented function, with lambda list (INSTANCE).

### **8.1263.2 File**

Defined in file src/db/friendly.lisp.

### **8.1263.3 SetF Function**

(SETF Toot-Pattern-Color) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1263.4 File**

Defined in file src/db/friendly.lisp.

## **8.1264 Tootsville::Toot-Pattern-Color-Name**

### **8.1264.1 Type**

Toot-Pattern-Color-Name names a TYPE:

The name of a color that can be used for a pattern

## 8.1265 Tootsville::Toot-Pattern-Color-Name-P

### 8.1265.1 Function

Toot-Pattern-Color-Name-P names an undocumented function, with lambda list (&REST ARGUMENTS).

### 8.1265.2 File

Defined in file `quicklisp/dists/quicklisp/software/fare-memoization-20180430-git/memoization.lisp`.

## **8.1266 Tootsville::Toot-Pattern-Name**

### **8.1266.1 Type**

Toot-Pattern-Name names a TYPE:

The name of a Toot pattern

## 8.1267 Tootsville::Toot-Pattern-Name-P

### 8.1267.1 Function

Toot-Pattern-Name-P names an undocumented function, with lambda list (&REST ARGUMENTS).

### 8.1267.2 File

Defined in file `quicklisp/dists/quicklisp/software/fare-memoization-20180430-git/memoization.lisp`.



## 8.1268 Tootsville::Toot-Peanuts

### 8.1268.1 Function

Toot-Peanuts names a function, with lambda list (TOOT):

Compute the total balance of peanuts that TOOT has earned over the course of the game.

### 8.1268.2 File

Defined in file src/quaestor.lisp.

## **8.1269 Tootsville::Toot-Personality**

### **8.1269.1 Class**

Toot-Personality names a class, with one superclass: Section 8.1072 [TOOTSVILLE ROBOT], page 1357.

### **8.1269.2 Slots**

Class Toot-Personality has no direct slots defined.

## 8.1270 Tootsville::Toot-Player

### 8.1270.1 Function

Toot-Player names an undocumented function, with lambda list (INSTANCE).

### 8.1270.2 File

Defined in file src/db/friendly.lisp.

### 8.1270.3 SetF Function

(SETF Toot-Player) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1270.4 File

Defined in file src/db/friendly.lisp.

## **8.1271 Tootsville::Toot-Position**

### **8.1271.1 Function**

Toot-Position names a function, with lambda list (TOOT):

Return the current point position of TOOT.

TOOT may be a Toot, robot, &c. The position returned will be a list of world keyword, latitude, longitude, and altitude.

### **8.1271.2 File**

Defined in file src/websockets.lisp.

### **8.1271.3 SetF Function**

(SETF Toot-Position) names an undocumented function, with lambda list (NEW-VALUE OBJECT).

## 8.1272 Tootsville::Toot-Presentation-Name

### 8.1272.1 Function

Toot-Presentation-Name names a function, with lambda list (TOOT):

The form of the TOOT's name for display in the UI as an avatar label.

This is usually the same as Section 8.1254 [TOOTSVILLE TOOT-NAME], page 1541, except for children or sensitive players, in which case it will have a black diamond prefixed to it.

### 8.1272.2 File

Defined in file src/toots.lisp.

## **8.1273 Tootsville::Toot-Private-Message**

### **8.1273.1 Function**

Toot-Private-Message names a function, with lambda list (SPEAKER LISTENER SPEECH):

SPEAKER whispers the message SPEECH to LISTENER.

### **8.1273.2 File**

Defined in file src/websockets.lisp.

## 8.1274 Tootsville::Toot-Quiesced

### 8.1274.1 Class

Toot-Quiesced names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.1274.2 Slots

Class Toot-Quiesced has 11 direct slot definitions:

Toot

World

Latitude

Longitude

Altitude

Wt1

D3

Emotion

Observed

Peer-Address

Attribs

## **8.1275 Tootsville::Toot-Quiesced-Altitude**

### **8.1275.1 Function**

Toot-Quiesced-Altitude names an undocumented function, with lambda list (INSTANCE).

### **8.1275.2 File**

Defined in file src/db/friendly.lisp.

### **8.1275.3 SetF Function**

(SETF Toot-Quiesced-Altitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1275.4 File**

Defined in file src/db/friendly.lisp.



## **8.1276 Tootsville::Toot-Quiesced-Attribs**

### **8.1276.1 Function**

Toot-Quiesced-Attribs names an undocumented function, with lambda list (INSTANCE).

### **8.1276.2 File**

Defined in file src/db/friendly.lisp.

### **8.1276.3 SetF Function**

(SETF Toot-Quiesced-Attribs) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1276.4 File**

Defined in file src/db/friendly.lisp.

## **8.1277 Tootsville::Toot-Quiesced-D3**

### **8.1277.1 Function**

Toot-Quiesced-D3 names an undocumented function, with lambda list (INSTANCE).

### **8.1277.2 File**

Defined in file src/db/friendly.lisp.

### **8.1277.3 SetF Function**

(SETF Toot-Quiesced-D3) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1277.4 File**

Defined in file src/db/friendly.lisp.

## 8.1278 Tootsville::Toot-Quiesced-Data

### 8.1278.1 Function

Toot-Quiesced-Data names an undocumented function, with lambda list (TOOT).

### 8.1278.2 File

Defined in file `src/characters/robots.lisp`.

## **8.1279 Tootsville::Toot-Quiesced-Emotion**

### **8.1279.1 Function**

Toot-Quiesced-Emotion names an undocumented function, with lambda list (INSTANCE).

### **8.1279.2 File**

Defined in file src/db/friendly.lisp.

### **8.1279.3 SetF Function**

(SETF Toot-Quiesced-Emotion) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1279.4 File**

Defined in file src/db/friendly.lisp.

## 8.1280 Tootsville::Toot-Quiesced-Latitude

### 8.1280.1 Function

Toot-Quiesced-Latitude names an undocumented function, with lambda list (INSTANCE).

### 8.1280.2 File

Defined in file src/db/friendly.lisp.

### 8.1280.3 SetF Function

(SETF Toot-Quiesced-Latitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1280.4 File

Defined in file src/db/friendly.lisp.

## 8.1281 Tootsville::Toot-Quiesced-Longitude

### 8.1281.1 Function

Toot-Quiesced-Longitude names an undocumented function, with lambda list (INSTANCE).

### 8.1281.2 File

Defined in file src/db/friendly.lisp.

### 8.1281.3 SetF Function

(SETF Toot-Quiesced-Longitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1281.4 File

Defined in file src/db/friendly.lisp.

## 8.1282 Tootsville::Toot-Quiesced-Observed

### 8.1282.1 Function

Toot-Quiesced-Observed names an undocumented function, with lambda list (INSTANCE).

### 8.1282.2 File

Defined in file src/db/friendly.lisp.

### 8.1282.3 SetF Function

(SETF Toot-Quiesced-Observed) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1282.4 File

Defined in file src/db/friendly.lisp.

## **8.1283 Tootsville::Toot-Quiesced-P**

### **8.1283.1 Function**

Toot-Quiesced-P names an undocumented function, with lambda list (OBJECT).

### **8.1283.2 File**

Defined in file src/db/friendly.lisp.



## 8.1284 Tootsville::Toot-Quiesced-Peer-Address

### 8.1284.1 Function

Toot-Quiesced-Peer-Address names an undocumented function, with lambda list (INSTANCE).

### 8.1284.2 File

Defined in file src/db/friendly.lisp.

### 8.1284.3 SetF Function

(SETF Toot-Quiesced-Peer-Address) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1284.4 File

Defined in file src/db/friendly.lisp.

## **8.1285 Tootsville::Toot-Quiesced-Toot**

### **8.1285.1 Function**

Toot-Quiesced-Toot names an undocumented function, with lambda list (INSTANCE).

### **8.1285.2 File**

Defined in file src/db/friendly.lisp.

### **8.1285.3 SetF Function**

(SETF Toot-Quiesced-Toot) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1285.4 File**

Defined in file src/db/friendly.lisp.

## **8.1286 Tootsville::Toot-Quiesced-World**

### **8.1286.1 Function**

Toot-Quiesced-World names an undocumented function, with lambda list (INSTANCE).

### **8.1286.2 File**

Defined in file src/db/friendly.lisp.

### **8.1286.3 SetF Function**

(SETF Toot-Quiesced-World) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1286.4 File**

Defined in file src/db/friendly.lisp.

## **8.1287 Tootsville::Toot-Quiesced-Wtl**

### **8.1287.1 Function**

Toot-Quiesced-Wtl names an undocumented function, with lambda list (INSTANCE).

### **8.1287.2 File**

Defined in file src/db/friendly.lisp.

### **8.1287.3 SetF Function**

(SETF Toot-Quiesced-Wtl) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1287.4 File**

Defined in file src/db/friendly.lisp.

## 8.1288 Tootsville::Toot-Sms-Messages

### 8.1288.1 Function

Toot-Sms-Messages names a function, with lambda list (TOOT &KEY (FROM NIL) (LIMIT 100)):

Find TOOT's SMS message starting with FROM, up to LIMIT.

FROM can be a UUID or an index from 0.

### 8.1288.2 File

Defined in file src/sms.lisp.

## **8.1289 Tootsville::Toot-Speak**

### **8.1289.1 Function**

Toot-Speak names a function, with lambda list (SPEECH &KEY (TOOT \*TOOT\*) VOL):

Broadcast a public message of SPEECH from TOOT at volume VOL.

### **8.1289.2 File**

Defined in file src/websockets.lisp.

## **8.1290 Tootsville::Toot-Uuid**

### **8.1290.1 Function**

Toot-Uuid names an undocumented function, with lambda list (INSTANCE).

### **8.1290.2 File**

Defined in file src/db/friendly.lisp.

### **8.1290.3 SetF Function**

(SETF Toot-Uuid) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1290.4 File**

Defined in file src/db/friendly.lisp.

## **8.1291 Tootsville::Tootsville-Rest-Acceptor**

### **8.1291.1 Class**

Tootsville-Rest-Acceptor names a class, with one superclass: HUNCHENTOOT::EASY-ACCEPTOR (not in this manual).

### **8.1291.2 Slots**

Class Tootsville-Rest-Acceptor has no direct slots defined.



## **8.1292 Tootsville::Tootsville-Rest-Ssl-Acceptor**

### **8.1292.1 Class**

Tootsville-Rest-Ssl-Acceptor names a class, with one superclass: HUNCHENTOOT::EASY-SSL-ACCEPTOR (not in this manual).

### **8.1292.2 Slots**

Class Tootsville-Rest-Ssl-Acceptor has no direct slots defined.

## **8.1293 Tootsville::Tootsville-V-Banner**

### **8.1293.1 Function**

Tootsville-V-Banner names an undocumented function, with lambda list NIL.

### **8.1293.2 File**

Defined in file src/version.lisp.

## **8.1294 Tootsville::Trace-Log-File**

### **8.1294.1 Function**

Trace-Log-File names a function, with lambda list (LOG-DIR):

Get the pathname of the trace log file.

### **8.1294.2 File**

Defined in file src/logging.lisp.

## **8.1295 Tootsville::Trace-Output-Heartbeat**

### **8.1295.1 Function**

Trace-Output-Heartbeat names a function, with lambda list NIL:

Output a heartbeat message with thread listing to `*TRACE-OUTPUT*` (see the Common Lisp HyperSpec)

### **8.1295.2 File**

Defined in file `src/logging.lisp`.

## **8.1296 Tootsville::Try-Reconnect-Toot-Name**

### **8.1296.1 Function**

Try-Reconnect-Toot-Name names a function, with lambda list (TOOT-NAME USER):

Allow TOOT-NAME to try to reconnect as USER.

### **8.1296.2 File**

Defined in file src/websockets.lisp.

## **8.1297 Tootsville::Two-Chars-In-A-Row-P**

### **8.1297.1 Function**

Two-Chars-In-A-Row-P names a function, with lambda list (STRING CHAR-BAG):

Do any two characters in CHAR-BAG occur together in STRING?

### **8.1297.2 File**

Defined in file `src/types/string-characteristics.lisp`.

## **8.1298 Tootsville::Two-Letter-String**

### **8.1298.1 Type**

Two-Letter-String names a TYPE:

A string of two letters (alphabetical characters)

## **8.1299 Tootsville::Un-Banhammer-Ip-Address**

### **8.1299.1 Function**

Un-Banhammer-Ip-Address names an undocumented function, with lambda list (ADDRESS).

### **8.1299.2 File**

Defined in file src/infinity/legacy-ops.lisp.



## 8.1300 Tootsville::Unicast

### 8.1300.1 Function

Unicast names a function, with lambda list (MESSAGE &OPTIONAL (USER (ACTIVE-PLAYER))):

Send MESSAGE directly to USER (which may be a Person or Toot)

### 8.1300.2 File

Defined in file src/messaging.lisp.

## **8.1301 Tootsville::Unidentified-Player-Error**

### **8.1301.1 Class**

Unidentified-Player-Error names a class, with one superclass: Section 8.667 [TOOTSVILLE HTTP-CLIENT-ERROR], page 925.

An error thrown when the player can't be identified.

They may have sent no credentials, or bad credentials.

### **8.1301.2 Slots**

Class Unidentified-Player-Error has 1 direct slot definition:

`Http-Status-Code`

## 8.1302 Tootsville::Unimplemented

### 8.1302.1 Class

Unimplemented names a class, with one superclass: Section 8.667 [TOOTSVILLE HTTP-CLIENT-ERROR], page 925.

Signals that a feature has not been implemented yet.

### 8.1302.2 Slots

Class Unimplemented has 2 direct slot definitions:

Http-Status-Code  
Feature

## **8.1303 Tootsville::Unimplemented-Feature**

### **8.1303.1 Function**

Unimplemented-Feature names an undocumented function, with lambda list (CONDITION).

### **8.1303.2 SetF Function**

(SETF Unimplemented-Feature) names an undocumented function, with lambda list (NEW-VALUE CONDITION).

## 8.1304 Tootsville::Unprocessable

### 8.1304.1 Class

Unprocessable names a class, with one superclass: Section 8.152 [TOOTSVILLE BAD-REQUEST], page 406.

A value submitted could not be processed.

### 8.1304.2 Slots

Class Unprocessable has 2 direct slot definitions:

`Http-Status-Code`  
`Thing`

## **8.1305 Tootsville::Update-Gravatar**

### **8.1305.1 Function**

Update-Gravatar names an undocumented function, with lambda list (PERSON EMAIL).

### **8.1305.2 File**

Defined in file src/users.lisp.

## **8.1306 Tootsville::Update-Nil**

### **8.1306.1 Class**

Update-Nil names a class, with one superclass: COMMON-LISP::CONDITION (not in this manual).

### **8.1306.2 Slots**

Class Update-Nil has no direct slots defined.

## **8.1307 Tootsville::Update-Toot-Last-Active**

### **8.1307.1 Function**

Update-Toot-Last-Active names a function, with lambda list (TOOT):

Set the Section 8.1252 [TOOTSVILLE TOOT-LAST-ACTIVE], page 1539, time for TOOT to the present time.

### **8.1307.2 File**

Defined in file `src/infinity/new-commands-20.lisp`.



## **8.1308 Tootsville::Uri-To-Uuid**

### **8.1308.1 Function**

Uri-To-Uuid names a function, with lambda list (UUID):

Extract a UUID encoded in Base64 in URI form.

### **8.1308.2 File**

Defined in file src/types/binary.lisp.

## **8.1309 Tootsville::Url-To-String**

### **8.1309.1 Function**

Url-To-String names a function, with lambda list (URL):

Converts URL to a string, if it is not already.

### **8.1309.2 File**

Defined in file src/users.lisp.

## **8.1310 Tootsville::User->Alist**

### **8.1310.1 Function**

User->Alist names an undocumented function, with lambda list (USER).

### **8.1310.2 File**

Defined in file src/users.lisp.

## **8.1311 Tootsville::User-Account**

### **8.1311.1 Function**

User-Account names an undocumented function, with lambda list (OBJECT).

### **8.1311.2 SetF Function**

(SETF User-Account) names an undocumented function, with lambda list (NEW-VALUE OBJECT).

## **8.1312 Tootsville::User-Display-Name**

### **8.1312.1 Function**

User-Display-Name names an undocumented function, with lambda list (&OPTIONAL (PERSON \*USER\*)).

### **8.1312.2 File**

Defined in file src/users.lisp.

## **8.1313 Tootsville::User-Email**

### **8.1313.1 Function**

User-Email names a function, with lambda list (&OPTIONAL (PERSON \*USER\*)):

Finds an email address for PERSON of type CONTACT.

### **8.1313.2 File**

Defined in file src/users.lisp.

## **8.1314 Tootsville::User-Face**

### **8.1314.1 Function**

User-Face names a function, with lambda list (&OPTIONAL (PERSON \*USER\*)):

    Finds a portrait URI for PERSON

### **8.1314.2 File**

Defined in file src/users.lisp.

## **8.1315 Tootsville::User-Given-Name**

### **8.1315.1 Function**

User-Given-Name names an undocumented function, with lambda list (&OPTIONAL (PERSON \*USER\*)).

### **8.1315.2 File**

Defined in file src/users.lisp.



## **8.1316 Tootsville::User-Id**

### **8.1316.1 Function**

User-Id names an undocumented function, with lambda list (&OPTIONAL (PERSON \*USER\*)).

### **8.1316.2 File**

Defined in file src/users.lisp.

## **8.1317 Tootsville::User-Online-P**

### **8.1317.1 Function**

User-Online-P names a function, with lambda list (USER):

Is USER actively connected right now?

### **8.1317.2 File**

Defined in file src/websockets.lisp.

## 8.1318 Tootsville::User-Stream

### 8.1318.1 Function

User-Stream names a function, with lambda list (WHOM):

Get the stream associated with WHOM.

WHOM might be a Toot, person, websocket client, robot, &c.

### 8.1318.2 File

Defined in file src/websockets.lisp.

## **8.1319 Tootsville::User-Surname**

### **8.1319.1 Function**

User-Surname names an undocumented function, with lambda list (&OPTIONAL (PERSON \*USER\*)).

### **8.1319.2 File**

Defined in file src/users.lisp.

## 8.1320 Tootsville::Uuid-String-P

### 8.1320.1 Function

Uuid-String-P names a function, with lambda list (STRING):

Does STRING look like a UUID?

Checks for 36 characters with #- in the correct positions and hex characters elsewhere.

### 8.1320.2 Example

6D559B46-D021-4814-A7F7-D8D67AD64800

### 8.1320.3 File

Defined in file src/types/string-characteristics.lisp.

## **8.1321 Tootsville::Uuid-String-To-Base64**

### **8.1321.1 Function**

Uuid-String-To-Base64 names a function, with lambda list (UUID-STRING):

Converts UUID-STRING into a UUID and gives its Base64 string value.

See also Section 8.1322 [TOOTSVILLE UUID-TO-BASE64], page 1609.

### **8.1321.2 File**

Defined in file src/db/db-central.lisp.

## 8.1322 Tootsville::Uuid-To-Base64

### 8.1322.1 Function

Uuid-To-Base64 names a function, with lambda list (UUID):

Convert UUID into a Base64 string.

Strips the trailing == that in invariant.

### 8.1322.2 File

Defined in file src/db/db-central.lisp.

## **8.1323 Tootsville::Uuid-To-Uri**

### **8.1323.1 Function**

Uuid-To-Uri names a function, with lambda list (UUID):

Encode UUID in Base64 and escape for URIs.

Swaps / characters for - characters to be more polite in an URI.

### **8.1323.2 File**

Defined in file src/types/binary.lisp.



## 8.1324 Tootsville::Valid-Child-Code-P

### 8.1324.1 Function

Valid-Child-Code-P names a function, with lambda list (CODE):

Is CODE valid for a child code?

It must be made up completely of ASCII67 characters and be 6-12 characters in length (inclusive).

### 8.1324.2 File

Defined in file `src/types/toot-names.lisp`.

## **8.1325 Tootsville::Value-To-Texi**

### **8.1325.1 Function**

Value-To-Texi names a function, with lambda list (SYMBOL):

Pretty-print the value of SYMBOL to a string.

Used for values of constants or default values of global (dynamic) variables.

### **8.1325.2 File**

Defined in file src/write-docs-2.lisp.

## **8.1326 Tootsville::Vanish-Item**

### **8.1326.1 Function**

Vanish-Item names a function, with lambda list (ITEM):

ITEM ceases to exist.

### **8.1326.2 File**

Defined in file src/items.lisp.

## **8.1327 Tootsville::Verbose-Log-File**

### **8.1327.1 Function**

Verbose-Log-File names a function, with lambda list (LOG-DIR):

Get the pathname of the verbose log file.

### **8.1327.2 File**

Defined in file src/logging.lisp.

## **8.1328 Tootsville::Version-Info-For**

### **8.1328.1 Function**

Version-Info-For names an undocumented function, with lambda list (ARGS).

### **8.1328.2 File**

Defined in file src/version.lisp.

## **8.1329 Tootsville::Version-Info-List**

### **8.1329.1 Function**

Version-Info-List names an undocumented function, with lambda list NIL.

### **8.1329.2 File**

Defined in file src/version.lisp.

## **8.1330 Tootsville::Version-Info-Report**

### **8.1330.1 Function**

Version-Info-Report names an undocumented function, with lambda list (&OPTIONAL (ARGS (QUOTE (\*)))).

### **8.1330.2 File**

Defined in file src/version.lisp.

## **8.1331 Tootsville::Version-Info-Report-String**

### **8.1331.1 Function**

Version-Info-Report-String names an undocumented function, with lambda list (ARGS).

### **8.1331.2 File**

Defined in file src/version.lisp.



## 8.1332 Tootsville::Wallet-Info

### 8.1332.1 Function

Wallet-Info names a function, with lambda list (TOOT):

Returns JSON-type data about TOOT's wallet.

This object contains

`walletOwner`

The Toot name whose wallet is being described

`currency` An object containing an enumeration of currencies. Each key is a currency's ISO symbol; each value is the amount of that currency which TOOT currently possesses.

### 8.1332.2 Changes from 1.2 to 2.0

In 1.2, the only currency reported was X-TVPM, Tootsville Magic Peanuts. Now, we also report (at least) X-FADU, fairy dust.

### 8.1332.3 File

Defined in file `src/toots.lisp`.

## **8.1333 Tootsville::Wants-Json-P**

### **8.1333.1 Function**

Wants-Json-P names a function, with lambda list NIL:

Does the client request Accept JSON format?

Looks for the canonical "Accept: application/json", and also checks the request URI for ".js" (which is, of course, a subseq of ".json" as well.)

### **8.1333.2 File**

Defined in file src/web.lisp.

## 8.1334 Tootsville::Wear-Slot

### 8.1334.1 Class

Wear-Slot names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.1334.2 Slots

Class Wear-Slot has 8 direct slot definitions:

Id

Name

Alternate

Avatar-Point

Valence

Obstruct-Point

Obstruct-Min

Obstruct-Max

## **8.1335 Tootsville::Wear-Slot-Alternate**

### **8.1335.1 Function**

Wear-Slot-Alternate names an undocumented function, with lambda list (INSTANCE).

### **8.1335.2 File**

Defined in file src/db/friendly.lisp.

### **8.1335.3 SetF Function**

(SETF Wear-Slot-Alternate) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1335.4 File**

Defined in file src/db/friendly.lisp.

## **8.1336 Tootsville::Wear-Slot-Avatar-Point**

### **8.1336.1 Function**

Wear-Slot-Avatar-Point names an undocumented function, with lambda list (INSTANCE).

### **8.1336.2 File**

Defined in file src/db/friendly.lisp.

### **8.1336.3 SetF Function**

(SETF Wear-Slot-Avatar-Point) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1336.4 File**

Defined in file src/db/friendly.lisp.

## **8.1337 Tootsville::Wear-Slot-Id**

### **8.1337.1 Function**

Wear-Slot-Id names an undocumented function, with lambda list (INSTANCE).

### **8.1337.2 File**

Defined in file src/db/friendly.lisp.

### **8.1337.3 SetF Function**

(SETF Wear-Slot-Id) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1337.4 File**

Defined in file src/db/friendly.lisp.

## 8.1338 Tootsville::Wear-Slot-Info

### 8.1338.1 Function

Wear-Slot-Info names a function, with lambda list (WEAR-SLOT):

Provides a JSON-style Plist describing WEAR-SLOT.

<b>id</b>	The unique ID of this wear-slot.						
<b>name</b>	The (potentially user-visible) name of this wear-slot.						
<b>alternate</b>	If this wear-slot has an alternate slot associated with it, this will be the wear-slot-ID of the alternate slot.						
<b>avatarPoint</b>	The moniker of the point on the avatar to which an item in this slot is mounted.						
<b>valence</b>	The valence level of this wear-slot on that avatarPoint. Multiple items mounted on one wear-slot can exist in valence levels.						
<b>obstruct</b>	If wearing an item in this slot obstructs the character from also wearing items in certain other slots: <table><tr><td><b>point</b></td><td>The <b>avatarPoint</b> which is obstructed,</td></tr><tr><td><b>min</b></td><td>The minimum valence level obstructed,</td></tr><tr><td><b>max</b></td><td>and the maximum valence level obstructed.</td></tr></table>	<b>point</b>	The <b>avatarPoint</b> which is obstructed,	<b>min</b>	The minimum valence level obstructed,	<b>max</b>	and the maximum valence level obstructed.
<b>point</b>	The <b>avatarPoint</b> which is obstructed,						
<b>min</b>	The minimum valence level obstructed,						
<b>max</b>	and the maximum valence level obstructed.						

### 8.1338.2 File

Defined in file src/items.lisp.

## **8.1339 Tootsville::Wear-Slot-Name**

### **8.1339.1 Function**

Wear-Slot-Name names an undocumented function, with lambda list (INSTANCE).

### **8.1339.2 File**

Defined in file src/db/friendly.lisp.

### **8.1339.3 SetF Function**

(SETF Wear-Slot-Name) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1339.4 File**

Defined in file src/db/friendly.lisp.



## **8.1340 Tootsville::Wear-Slot-Obstruct-Max**

### **8.1340.1 Function**

Wear-Slot-Obstruct-Max names an undocumented function, with lambda list (INSTANCE).

### **8.1340.2 File**

Defined in file src/db/friendly.lisp.

### **8.1340.3 SetF Function**

(SETF Wear-Slot-Obstruct-Max) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1340.4 File**

Defined in file src/db/friendly.lisp.

## **8.1341 Tootsville::Wear-Slot-Obstruct-Min**

### **8.1341.1 Function**

Wear-Slot-Obstruct-Min names an undocumented function, with lambda list (INSTANCE).

### **8.1341.2 File**

Defined in file src/db/friendly.lisp.

### **8.1341.3 SetF Function**

(SETF Wear-Slot-Obstruct-Min) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1341.4 File**

Defined in file src/db/friendly.lisp.

## 8.1342 Tootsville::Wear-Slot-Obstruct-Point

### 8.1342.1 Function

Wear-Slot-Obstruct-Point names an undocumented function, with lambda list (INSTANCE).

### 8.1342.2 File

Defined in file src/db/friendly.lisp.

### 8.1342.3 SetF Function

(SETF Wear-Slot-Obstruct-Point) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1342.4 File

Defined in file src/db/friendly.lisp.

## **8.1343 Tootsville::Wear-Slot-P**

### **8.1343.1 Function**

Wear-Slot-P names an undocumented function, with lambda list (OBJECT).

### **8.1343.2 File**

Defined in file src/db/friendly.lisp.

## **8.1344 Tootsville::Wear-Slot-Valence**

### **8.1344.1 Function**

Wear-Slot-Valence names an undocumented function, with lambda list (INSTANCE).

### **8.1344.2 File**

Defined in file src/db/friendly.lisp.

### **8.1344.3 SetF Function**

(SETF Wear-Slot-Valence) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1344.4 File**

Defined in file src/db/friendly.lisp.

## **8.1345 Tootsville::Websocket-Acceptor**

### **8.1345.1 Class**

Websocket-Acceptor names a class, with one superclass: HUNCHENSOCKET::WEBSOCKET-ACCEPTOR (not in this manual).

### **8.1345.2 Slots**

Class Websocket-Acceptor has no direct slots defined.

## 8.1346 Tootsville::Websocket-Authenticate

### 8.1346.1 Function

Websocket-Authenticate names a function, with lambda list (CLIENT AUTH\$):

CLIENT wishes to authenticate using AUTH\$, a string containing JSON data.

AUTH\$ must be a packet in one of the following forms:

- It may be a direct login using a known authentication provider, in which case it will contain a key `Auth/∞/ℕ` (that is, auth infinity alef-null) and be passed to Section 8.586 [TOOTSVILLE FIND-USER-FOR-JSON], page 843, for processing.
- It may be a Toot-based login (now for children only) and send a `getApple` request followed by a `login` request. These are handled by Section 8.693 [TOOTSVILLE INFINITY-GET-APPLE], page 956, (qv for details of this mechanism) and Section 8.713 [TOOTSVILLE INFINITY-LOGIN], page 983.
- For compatibility, a few other packet types may be ignored by this function but are no longer processed. See Section 8.719 [TOOTSVILLE INFINITY-PRE-LOGIN], page 990, for details.

The client is required to sign in within a few seconds and can issue no more than a few commands before being dropped. See Section 8.99 [TOOTSVILLE +PRE-LOGIN-MAX-TIME+], page 353, and Section 8.98 [TOOTSVILLE +PRE-LOGIN-MAX-COMMANDS+], page 352.

### 8.1346.2 File

Defined in file `src/websockets.lisp`.

## 8.1347 Tootsville::Websocket-Ssl-Acceptor

### 8.1347.1 Class

Websocket-Ssl-Acceptor names a class, with one superclass: HUNCHENSOCKET::WEBSOCKET-SSL-ACCEPTOR (not in this manual).

### 8.1347.2 Slots

Class Websocket-Ssl-Acceptor has no direct slots defined.



## 8.1348 Tootsville::Which-Toot-Is-Not-Yours

### 8.1348.1 Function

Which-Toot-Is-Not-Yours names an undocumented function, with lambda list (CONDITION).

### 8.1348.2 SetF Function

(SETF Which-Toot-Is-Not-Yours) names an undocumented function, with lambda list (NEW-VALUE CONDITION).

## **8.1349 Tootsville::Whitespace-Char-P**

### **8.1349.1 Function**

Whitespace-Char-P names an undocumented function, with lambda list (CHARACTER).

### **8.1349.2 File**

Defined in file src/acceptor.lisp.

## **8.1350 Tootsville::Whitespacep**

### **8.1350.1 Function**

Whitespacep names an undocumented function, with lambda list (STRING).

### **8.1350.2 File**

Defined in file src/acceptor.lisp.

## **8.1351 Tootsville::Who-Is-Connected**

### **8.1351.1 Function**

Who-Is-Connected names a function, with lambda list NIL:

All users currently connected via websockets.

Returns person objects, removing nulls for unauthenticated users.

### **8.1351.2 File**

Defined in file src/websockets.lisp.

## 8.1352 Tootsville::Wind-Vector

### 8.1352.1 Class

Wind-Vector names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.1352.2 Slots

Class Wind-Vector has 2 direct slot definitions:

X-Magnitude

Y-Magnitude

## **8.1353 Tootsville::Wind-Vector-P**

### **8.1353.1 Function**

Wind-Vector-P names an undocumented function, with lambda list (OBJECT).

### **8.1353.2 File**

Defined in file src/weather/weather.lisp.

## **8.1354 Tootsville::Wind-Vector-X-Magnitude**

### **8.1354.1 Function**

Wind-Vector-X-Magnitude names an undocumented function, with lambda list (INSTANCE).

### **8.1354.2 File**

Defined in file src/weather/weather.lisp.

### **8.1354.3 SetF Function**

(SETF Wind-Vector-X-Magnitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1354.4 File**

Defined in file src/weather/weather.lisp.

## **8.1355 Tootsville::Wind-Vector-Y-Magnitude**

### **8.1355.1 Function**

Wind-Vector-Y-Magnitude names an undocumented function, with lambda list (INSTANCE).

### **8.1355.2 File**

Defined in file src/weather/weather.lisp.

### **8.1355.3 SetF Function**

(SETF Wind-Vector-Y-Magnitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1355.4 File**

Defined in file src/weather/weather.lisp.



## **8.1356 Tootsville::Wind-X**

### **8.1356.1 Function**

Wind-X names a function, with lambda list (WIND-VECTOR):  
The X component of WIND-VECTOR.

### **8.1356.2 File**

Defined in file src/weather/weather.lisp.

## **8.1357 Tootsville::Wind-Y**

### **8.1357.1 Function**

Wind-Y names a function, with lambda list (WIND-VECTOR):  
The Y component of WIND-VECTOR.

### **8.1357.2 File**

Defined in file src/weather/weather.lisp.

## 8.1358 Tootsville::With-Cluster-Wide-Lock-Held

### 8.1358.1 Macro

With-Cluster-Wide-Lock-Held names a macro, with lambda list ((LOCK-STRING &KEY (TIMEOUT) (IF-NOT-LOCKED)) &BODY BODY):

Execute BODY in a dynamic context owning database lock LOCK-STRING.

LOCK-STRING is passed to the MariaDB server and a global lock by that name is obtained via MySQL function GET\_LOCK(STRING), if possible.

If the lock is busy, IF-NOT-LOCKED determines the next action.

**:WAIT**

Wait for up to TIMEOUT seconds for the lock to be freed. If the lock cannot be obtained within TIMEOUT seconds, signal an error of type CLUSTER-WIDE-LOCK-BUSY-ERROR. If TIMEOUT is NIL, wait indefinitely until the lock can be obtained.

**:SKIP**

Skip BODY and return NIL.

**:WARN**

Signal a warning of type CLUSTER-WIDE-LOCK-BUSY-WARNING, then skip BODY and return NIL.

**:ERROR**

Signal an error of type CLUSTER-WIDE-LOCK-BUSY-ERROR.

Returns the values of BODY.

### 8.1358.2 File

Defined in file src/db/maria.lisp.

## **8.1359 Tootsville::With-Continuable-Errors-Skipped**

### **8.1359.1 Macro**

With-Continuable-Errors-Skipped names an undocumented macro, with lambda list (&BODY BODY).

### **8.1359.2 File**

Defined in file src/endpoints/slash-maintenance.lisp.

## **8.1360 Tootsville::With-Dbi**

### **8.1360.1 Macro**

With-Dbi names an undocumented macro, with lambda list ((MONIKER) &BODY BODY).

### **8.1360.2 File**

Defined in file src/db/maria.lisp.

## **8.1361 Tootsville::With-Errors-As-Http**

### **8.1361.1 Macro**

With-Errors-As-Http names a macro, with lambda list ((ERROR-CODE &OPTIONAL THING) &BODY BODY):

Execute BODY in a context in which any error results in HTTP ERROR-CODE.

Rather than defaulting to an HTTP 500, ERROR-CODE will be returned as the outcome of any uncaught error signal.

### **8.1361.2 File**

Defined in file src/web.lisp.

## **8.1362 Tootsville::With-Http-Conditions**

### **8.1362.1 Macro**

With-Http-Conditions names an undocumented macro, with lambda list (NIL &BODY BODY).

### **8.1362.2 File**

Defined in file src/acceptor.lisp.

## **8.1363 Tootsville::With-Http-Errors-As-Infinity-Errors**

### **8.1363.1 Macro**

With-Http-Errors-As-Infinity-Errors names an undocumented macro, with lambda list ((COMMAND) &BODY BODY).

### **8.1363.2 File**

Defined in file src/infinity/infinity.lisp.



## **8.1364 Tootsville::With-Local-Toot**

### **8.1364.1 Macro**

With-Local-Toot names a macro, with lambda list ((TOOT) &BODY BODY):

Set \*TOOT\* to the Toot named TOOT.

### **8.1364.2 File**

Defined in file src/users.lisp.

## **8.1365 Tootsville::With-Local-User**

### **8.1365.1 Macro**

With-Local-User names a macro, with lambda list ((EMAIL) &BODY BODY):

Set \*USER\* to the user with EMAIL locally

### **8.1365.2 File**

Defined in file src/users.lisp.

## 8.1366 Tootsville::With-Maintenance-Times

### 8.1366.1 Macro

With-Maintenance-Times names an undocumented macro, with lambda list ((TASK-NAME TASK-STRING START-DELAY FINISH-DELAY) &BODY BODY).

### 8.1366.2 File

Defined in file `src/endpoints/slash-maintenance.lisp`.

## **8.1367 Tootsville::With-Memcached-Query**

### **8.1367.1 Macro**

With-Memcached-Query names a macro, with lambda list ((DB QUERY ARGS &KEY (TIMEOUT)) &BODY BODY):

Execute BODY only if the QUERY's value is not found in MemCacheD.

### **8.1367.2 File**

Defined in file src/db/memcached.lisp.

## 8.1368 Tootsville::With-Posted-Json

### 8.1368.1 Macro

With-Posted-Json names a macro, with lambda list `((&REST A-LIST) &BODY BODY)`:

Execute `BODY` with `A-LIST` values from JSON body of a POST.

Each variable named in `A-LIST` will be bound to the `JONATHAN::PARSE` (not in this manual) contents of the analogous (camel-case) key name in the POSTed parameter object.

For example,

```
(WITH-POSTED-JSON (FOO-BAR)
 (BODY))
```

... will bind `FOO-BAR` to the value of the key `"fooBar"` in the POST content, assuming it is a JSON object like

```
{ "fooBar": "value" }
```

In the event of a parse error, an HTTP 400 is returned.

### 8.1368.2 File

Defined in file `src/web.lisp`.

## 8.1369 Tootsville::With-Score-In-Range

### 8.1369.1 Macro

With-Score-In-Range names a macro, with lambda list ((SCORE MIN &OPTIONAL MAX) &BODY BODY):

Assert that SCORE is in range of MIN (to MAX, if any) and run BODY, or return a score.range error.

### 8.1369.2 File

Defined in file src/quaestor.lisp.

## **8.1370 Tootsville::With-Standard-Streams-To-String**

### **8.1370.1 Macro**

With-Standard-Streams-To-String names an undocumented macro, with lambda list (&BODY BODY).

### **8.1370.2 File**

Defined in file `src/endpoints/slash-maintenance.lisp`.

## **8.1371 Tootsville::With-User**

### **8.1371.1 Macro**

With-User names an undocumented macro, with lambda list (NIL &BODY BODY).

### **8.1371.2 File**

Defined in file src/users.lisp.



## 8.1372 Tootsville::With-Websocket-Disconnections

### 8.1372.1 Macro

With-Websocket-Disconnections names a macro, with lambda list ((CLIENT) &BODY BODY):

Handle errors caused by surprise disconnections by CLIENT.

### 8.1372.2 File

Defined in file src/websockets.lisp.

## **8.1373 Tootsville::Without-Medal**

### **8.1373.1 Macro**

Without-Medal names a macro, with lambda list ((MEDAL) &BODY BODY):

Assert that MEDAL is null and run BODY, or return a medal.notFound error.

### **8.1373.2 File**

Defined in file src/quaestor.lisp.

## **8.1374 Tootsville::Without-Sem**

### **8.1374.1 Function**

Without-Sem names a function, with lambda list (STRING):

The subset of STRING up to the first semicolon, if any.

### **8.1374.2 File**

Defined in file src/web.lisp.

## **8.1375 Tootsville::World**

### **8.1375.1 Function**

World names a function, with lambda list (THING):

The keyword name of the world on which THING is.

### **8.1375.2 File**

Defined in file src/characters/robots.lisp.

### **8.1375.3 Class**

World names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### **8.1375.4 Slots**

Class World has 2 direct slot definitions:

Moniker

Name

## **8.1376 Tootsville::World-Mistp**

### **8.1376.1 Function**

World-Mistp names an undocumented function, with lambda list (LATITUDE LONGITUDE ALTITUDE WORLD).

### **8.1376.2 File**

Defined in file src/world.lisp.

## **8.1377 Tootsville::World-Moniker**

### **8.1377.1 Function**

World-Moniker names an undocumented function, with lambda list (INSTANCE).

### **8.1377.2 File**

Defined in file src/db/friendly.lisp.

### **8.1377.3 SetF Function**

(SETF World-Moniker) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1377.4 File**

Defined in file src/db/friendly.lisp.

### **8.1377.5 Type**

World-Moniker names a TYPE:

cv Section 8.891 [TOOTSVILLE MAP-PLACES], page 1176,

## 8.1378 Tootsville::World-Moniker-P

### 8.1378.1 Function

World-Moniker-P names a function, with lambda list (MONIKER):

cv. Section 8.891 [TOOTSVILLE MAP-PLACES], page 1176,

The monikers for the worlds are the hard list:

**CHOR** Choerogryllum (the planet on which Tootsville is found).

**MOON** The moon.

**OTHM** The other moon.

**PINK** The pink moon.

**ORBIT** In orbit of Choerogryllum, but not on any moon.

### 8.1378.2 File

Defined in file src/types/world-types.lisp.

## **8.1379 Tootsville::World-Name**

### **8.1379.1 Function**

World-Name names an undocumented function, with lambda list (INSTANCE).

### **8.1379.2 File**

Defined in file src/db/friendly.lisp.

### **8.1379.3 SetF Function**

(SETF World-Name) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1379.4 File**

Defined in file src/db/friendly.lisp.



## **8.1380 Tootsville::World-P**

### **8.1380.1 Function**

World-P names an undocumented function, with lambda list (OBJECT).

### **8.1380.2 File**

Defined in file src/db/friendly.lisp.

## **8.1381 Tootsville::Write-Class-Docs**

### **8.1381.1 Function**

Write-Class-Docs names a function, with lambda list (SYMBOL METAOBJECT S):

Write documentation for class named SYMBOL with metaobject METAOBJECT to stream S

### **8.1381.2 File**

Defined in file src/write-docs-2.lisp.

## 8.1382 Tootsville::Write-Docs

### 8.1382.1 Function

Write-Docs names a function, with lambda list NIL:

Write out the documentation in TeXinfo format.

XXX this is a huge function that ought to be broken up more

### 8.1382.2 File

Defined in file src/write-docs-2.lisp.

## **8.1383 Tootsville::Write-Docs-Header**

### **8.1383.1 Function**

Write-Docs-Header names an undocumented function, with lambda list (DOCS SOURCE-DIR).

### **8.1383.2 File**

Defined in file src/write-docs-2.lisp.

## **8.1384 Tootsville::Write-Documentation**

### **8.1384.1 Function**

Write-Documentation names a function, with lambda list (SYMBOL S):

Writes the TeXinfo documentation for SYMBOL to stream S.

Ignores “private” functions, indicated by a % in the first or last position of the name.

### **8.1384.2 File**

Defined in file src/write-docs-2.lisp.

## **8.1385 Tootsville::Write-Function-Docs**

### **8.1385.1 Function**

Write-Function-Docs names a function, with lambda list (SYMBOL S):

Write documentation for the function (or macro) SYMBOL to S

### **8.1385.2 File**

Defined in file src/write-docs-2.lisp.

## **8.1386 Tootsville::Write-Setf-Docs**

### **8.1386.1 Function**

Write-Setf-Docs names a function, with lambda list (SYMBOL S):

Write documentation for the SetF function (SETF SYMBOL) to S

### **8.1386.2 File**

Defined in file src/write-docs-2.lisp.

## **8.1387 Tootsville::Write-Staff-Journal-Entry**

### **8.1387.1 Function**

Write-Staff-Journal-Entry names a function, with lambda list (ENTRY WHO):

Write ENTRY to the staff journal, timestamped now. Reference WHO in the entry.

WHO may be anything accepted by Section 8.544 [TOOTSVILLE ENSURE-LIST-OF-PEOPLE], page 801.

Journal entries are associated with the person (people) owning the relevant Toot(s)

### **8.1387.2 File**

Defined in file src/staff-journal.lisp.



## **8.1388 Tootsville::Ws-Approve-Toot**

### **8.1388.1 Function**

Ws-Approve-Toot names a function, with lambda list (TOOT REQUEST):

Notify TOOT that REQUEST was approved.

REQUEST is a Section 8.200 [TOOTSVILLE CHILD-REQUEST], page 454,

### **8.1388.2 File**

Defined in file src/websockets.lisp.

## **8.1389 Tootsville::Ws-Bandwidth-By-Source**

### **8.1389.1 Function**

Ws-Bandwidth-By-Source names an undocumented function, with lambda list NIL.

### **8.1389.2 File**

Defined in file src/websockets.lisp.

## 8.1390 Tootsville::Ws-Bandwidth-Record

### 8.1390.1 Function

Ws-Bandwidth-Record names a function, with lambda list (PACKET &OPTIONAL (MULTIPLIER 1)):

Record bandwidth used by this PACKET.

For broadcasts, multiply by MULTIPLIER.

### 8.1390.2 File

Defined in file src/websockets.lisp.

## **8.1391 Tootsville::Ws-Broadcast**

### **8.1391.1 Function**

Ws-Broadcast names a function, with lambda list (RES MESSAGE &KEY NEAR EXCEPT):

Low-level broadcast MESSAGE to all WebSocket clients of resource RES near NEAR except EXCEPT.

You almost certainly don't want to call this — you want Section 8.166 [TOOTSVILLE BROADCAST], page 420.

### **8.1391.2 File**

Defined in file src/websockets.lisp.

## **8.1392 Tootsville::Ws-Client**

### **8.1392.1 Class**

Ws-Client names a class, with one superclass: HUNCHENSOCKET::WEBSOCKET-CLIENT (not in this manual).

### **8.1392.2 Slots**

Class Ws-Client has no direct slots defined.

## **8.1393 Tootsville::Ws-Deny-Toot**

### **8.1393.1 Function**

Ws-Deny-Toot names a function, with lambda list (TOOT REQUEST):

Notify TOOT that REQUEST was denied

REQUEST is a Section 8.200 [TOOTSVILLE CHILD-REQUEST], page 454,

### **8.1393.2 File**

Defined in file src/websockets.lisp.

## 8.1394 Tootsville::Ws-Evacuate-All

### 8.1394.1 Function

Ws-Evacuate-All names a function, with lambda list (&OPTIONAL (RESOURCE \*INFINITY-WEBSOCKET-RESOURCE\*)):

Evacuate all connected players to other servers.

Broadcasts a **from: "migrate"** packet to all users connected to RESOURCE.

### 8.1394.2 File

Defined in file src/websockets.lisp.

## **8.1395 Tootsville::Ws-Kick**

### **8.1395.1 Function**

Ws-Kick names a function, with lambda list (CLIENT):

Kick a WebSocket connected user off-line

### **8.1395.2 File**

Defined in file src/websockets.lisp.



## **8.1396 Tootsville::Ws-Kick-Other-Streams-For-User**

### **8.1396.1 Function**

Ws-Kick-Other-Streams-For-User names a function, with lambda list (&OPTIONAL (USER \*USER\*)):

Section 8.1395 [TOOTSVILLE WS-KICK], page 1682, any stream on which USER is signed in.

### **8.1396.2 File**

Defined in file src/websockets.lisp.

## **8.1397 Tootsville::Ws-Perform-Sign-In**

### **8.1397.1 Function**

Ws-Perform-Sign-In names a function, with lambda list (CLIENT &OPTIONAL (USER \*USER\*)):

Perform signing in USER on CLIENT and side-effects.

Calls Section 8.1399 [TOOTSVILLE WS-SIGN-IN-USER], page 1686, and Section 8.1396 [TOOTSVILLE WS-KICK-OTHER-STREAMS-FOR-USER], page 1683,

Sends logOK message and Toots List

### **8.1397.2 File**

Defined in file src/websockets.lisp.

## **8.1398 Tootsville::Ws-Reply**

### **8.1398.1 Function**

Ws-Reply names a function, with lambda list (MESSAGE WS-CLIENT):

Send a reply MESSAGE to a WebSocket WS-CLIENT from an Infinity handler.

### **8.1398.2 File**

Defined in file src/websockets.lisp.

## **8.1399 Tootsville::Ws-Sign-In-User**

### **8.1399.1 Function**

Ws-Sign-In-User names a function, with lambda list (CLIENT &OPTIONAL (USER \*USER\*)):

Sign in USER on CLIENT connection.

The full procedure comes about from Section 8.1397 [TOOTSVILLE WS-PERFORM-SIGN-IN], page 1684. This function only handles the low-level bookkeeping.

### **8.1399.2 File**

Defined in file src/websockets.lisp.

## **8.1400 Tootsville::Ws-Stats**

### **8.1400.1 Function**

Ws-Stats names a function, with lambda list NIL:

Returns a string with some nifty statistics about WebSockets

### **8.1400.2 File**

Defined in file src/websockets.lisp.

## **8.1401 Tootsville::Ws-Stats-Reset-All**

### **8.1401.1 Function**

Ws-Stats-Reset-All names an undocumented function, with lambda list NIL.

### **8.1401.2 File**

Defined in file src/websockets.lisp.

## **8.1402 Tootsville::Ws-To-Infinity**

### **8.1402.1 Function**

Ws-To-Infinity names an undocumented function, with lambda list (CLIENT MESSAGE).

### **8.1402.2 File**

Defined in file src/websockets.lisp.

## **8.1403 Tootsville::Ws-Uncast**

### **8.1403.1 Function**

Ws-Uncast names a function, with lambda list (MESSAGE USER):

Low-level unicast MESSAGE to USER over WebSockets

### **8.1403.2 File**

Defined in file src/websockets.lisp.



## **8.1404 Tootsville::Ws-Without-Login**

### **8.1404.1 Function**

Ws-Without-Login names an undocumented function, with lambda list (CLIENT MESSAGE).

### **8.1404.2 File**

Defined in file src/websockets.lisp.

## 8.1405 Tootsville::Wtl-Course

### 8.1405.1 Function

Wtl-Course names a function, with lambda list (THING):

The course of THING's current movement in WTL form.

See Section 8.744 [TOOTSVILLE INFINITY-WTL], page 1028, for a discussion of this format.

### 8.1405.2 File

Defined in file src/characters/robots.lisp.

### 8.1405.3 Class

Wtl-Course names a class, with one superclass: COMMON-LISP::STRUCTURE-OBJECT (not in this manual).

### 8.1405.4 Slots

Class Wtl-Course has 10 direct slot definitions:

Speed

Start-Time

End-Time

Start-Point

End-Point

Latitude

Longitude

Altitude

World

Facing

## 8.1406 Tootsville::Wtl-Course-Altitude

### 8.1406.1 Function

Wtl-Course-Altitude names an undocumented function, with lambda list (INSTANCE).

### 8.1406.2 File

Defined in file src/characters/robots.lisp.

### 8.1406.3 SetF Function

(SETF Wtl-Course-Altitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1406.4 File

Defined in file src/characters/robots.lisp.

## **8.1407 Tootsville::Wtl-Course-End-Point**

### **8.1407.1 Function**

Wtl-Course-End-Point names an undocumented function, with lambda list (INSTANCE).

### **8.1407.2 File**

Defined in file src/characters/robots.lisp.

### **8.1407.3 SetF Function**

(SETF Wtl-Course-End-Point) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1407.4 File**

Defined in file src/characters/robots.lisp.

## **8.1408 Tootsville::Wtl-Course-End-Time**

### **8.1408.1 Function**

Wtl-Course-End-Time names an undocumented function, with lambda list (INSTANCE).

### **8.1408.2 File**

Defined in file src/characters/robots.lisp.

### **8.1408.3 SetF Function**

(SETF Wtl-Course-End-Time) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1408.4 File**

Defined in file src/characters/robots.lisp.

## **8.1409 Tootsville::Wtl-Course-Facing**

### **8.1409.1 Function**

Wtl-Course-Facing names an undocumented function, with lambda list (INSTANCE).

### **8.1409.2 File**

Defined in file src/characters/robots.lisp.

### **8.1409.3 SetF Function**

(SETF Wtl-Course-Facing) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1409.4 File**

Defined in file src/characters/robots.lisp.

## 8.1410 Tootsville::Wtl-Course-Latitude

### 8.1410.1 Function

Wtl-Course-Latitude names an undocumented function, with lambda list (INSTANCE).

### 8.1410.2 File

Defined in file src/characters/robots.lisp.

### 8.1410.3 SetF Function

(SETF Wtl-Course-Latitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### 8.1410.4 File

Defined in file src/characters/robots.lisp.

## **8.1411 Tootsville::Wtl-Course-Longitude**

### **8.1411.1 Function**

Wtl-Course-Longitude names an undocumented function, with lambda list (INSTANCE).

### **8.1411.2 File**

Defined in file src/characters/robots.lisp.

### **8.1411.3 SetF Function**

(SETF Wtl-Course-Longitude) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1411.4 File**

Defined in file src/characters/robots.lisp.



## **8.1412 Tootsville::Wtl-Course-P**

### **8.1412.1 Function**

Wtl-Course-P names an undocumented function, with lambda list (OBJECT).

### **8.1412.2 File**

Defined in file `src/characters/robots.lisp`.

## **8.1413 Tootsville::Wtl-Course-Speed**

### **8.1413.1 Function**

Wtl-Course-Speed names an undocumented function, with lambda list (INSTANCE).

### **8.1413.2 File**

Defined in file src/characters/robots.lisp.

### **8.1413.3 SetF Function**

(SETF Wtl-Course-Speed) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1413.4 File**

Defined in file src/characters/robots.lisp.

## **8.1414 Tootsville::Wtl-Course-Start-Point**

### **8.1414.1 Function**

Wtl-Course-Start-Point names an undocumented function, with lambda list (INSTANCE).

### **8.1414.2 File**

Defined in file src/characters/robots.lisp.

### **8.1414.3 SetF Function**

(SETF Wtl-Course-Start-Point) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1414.4 File**

Defined in file src/characters/robots.lisp.

## **8.1415 Tootsville::Wtl-Course-Start-Time**

### **8.1415.1 Function**

Wtl-Course-Start-Time names an undocumented function, with lambda list (INSTANCE).

### **8.1415.2 File**

Defined in file src/characters/robots.lisp.

### **8.1415.3 SetF Function**

(SETF Wtl-Course-Start-Time) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1415.4 File**

Defined in file src/characters/robots.lisp.

## **8.1416 Tootsville::Wtl-Course-World**

### **8.1416.1 Function**

Wtl-Course-World names an undocumented function, with lambda list (INSTANCE).

### **8.1416.2 File**

Defined in file src/characters/robots.lisp.

### **8.1416.3 SetF Function**

(SETF Wtl-Course-World) names an undocumented function, with lambda list (VALUE INSTANCE).

### **8.1416.4 File**

Defined in file src/characters/robots.lisp.

## **8.1417 Tootsville::Www-Uri**

### **8.1417.1 Type**

Www-Uri names a TYPE:

A string that could be a WWW URI

See Section 8.1418 [TOOTSVILLE WWW-URI-LIKE-P], page 1705.

## **8.1418 Tootsville::Www-Uri-Like-P**

### **8.1418.1 Function**

Www-Uri-Like-P names a function, with lambda list (URI):

Does URI look like a WWW (HTTP/HTTPS) URI?

### **8.1418.2 File**

Defined in file src/types/uri-types.lisp.

## **8.1419 Tootsville::Yesterday**

### **8.1419.1 Function**

Yesterday names a function, with lambda list NIL:

Get a timestamp for yesterday.

### **8.1419.2 File**

Defined in file src/types/date+time.lisp.



## 8.1420 Tootsville::Yield-Mariadb-Lock

### 8.1420.1 Function

Yield-Mariadb-Lock names a function, with lambda list (LOCK-NAME):

Release the lock identified by LOCK-NAME.

LOCK-NAME is case-insensitive.

### 8.1420.2 File

Defined in file src/db/ maria.lisp.

## **8.1421 Tootsville::Zap-Personality**

### **8.1421.1 Class**

Zap-Personality names a class, with one superclass: Section 8.1107 [TOOTSVILLE ROBOT-ZAP], page 1392.

This class defines a character named Zap

### **8.1421.2 Slots**

Class Zap-Personality has no direct slots defined.

## 8.1422 Tootsville::⊕Post-Accept-Type-Does-Not-Match-/ \*-When-Not-Allow-Wildcards-P

### 8.1422.1 Function

⊕Post-Accept-Type-Does-Not-Match-/\*-When-Not-Allow-Wildcards-P names a function, with lambda list NIL:

The Accept: type with :ALLOW-WILDCARD-P NIL does not match a wildcard

### 8.1422.2 File

Defined in file src/acceptor.lisp.

## 8.1423 Tootsville::⊕Post-Accept-Type-Matches-\*/ \*

### 8.1423.1 Function

⊕Post-Accept-Type-Matches-\*/ \* names a function, with lambda list NIL:

The Accept: type must match \*/\*

### 8.1423.2 File

Defined in file src/acceptor.lisp.

## 8.1424 Tootsville::⊕Post-Accept-Type-Matches-/ \*

### 8.1424.1 Function

⊕Post-Accept-Type-Matches-/ \* names a function, with lambda list NIL:

The Accept: type must match a wildcard like text/\*

### 8.1424.2 File

Defined in file src/acceptor.lisp.

## 8.1425 Tootsville::⊕Post-Accept-Type-Matches-/\*-With-Charset=UTF-8

### 8.1425.1 Function

⊕Post-Accept-Type-Matches-/\*-With-Charset=UTF-8 names a function, with lambda list NIL:

The Accept: type must match a wildcard like text/\* with ;charset=utf-8

### 8.1425.2 File

Defined in file src/acceptor.lisp.

## 8.1426 Tootsville::⊕Post-Accept-Type-Matches-Identically

### 8.1426.1 Function

⊕Post-Accept-Type-Matches-Identically names a function, with lambda list NIL:

The Accept: type must match with exact matching.

### 8.1426.2 File

Defined in file src/acceptor.lisp.

## 8.1427 Tootsville::⊕Post-Accept-Type-Matches-With-Charset=Utf-8

### 8.1427.1 Function

⊕Post-Accept-Type-Matches-With-Charset=Utf-8 names a function, with lambda list NIL:

The Accept: type must match with trailing ;charset=utf-8

### 8.1427.2 File

Defined in file src/acceptor.lisp.



## 8.1428 Tootsville::⊕Post-Acceptor-Template-Matches-Constants

### 8.1428.1 Function

⊕Post-Acceptor-Template-Matches-Constants names a function, with lambda list NIL:

An acceptor template list must match constants.

### 8.1428.2 File

Defined in file src/acceptor.lisp.

## 8.1429 Tootsville::⊕Post-Acceptor-Template-Unifies-Variables

### 8.1429.1 Function

⊕Post-Acceptor-Template-Unifies-Variables names a function, with lambda list NIL:

An acceptor template list must match variables and return their bindings.

### 8.1429.2 File

Defined in file src/acceptor.lisp.

## 8.1430 Tootsville::⊕Post-Certificate-Extraction

### 8.1430.1 Function

⊕Post-Certificate-Extraction names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Certificate-Extraction.

### 8.1430.2 File

Defined in file src/auth/auth-firebase.lisp.

## 8.1431 Tootsville::⊕Post-Check-Map-Heights

### 8.1431.1 Function

⊕Post-Check-Map-Heights names a function, with lambda list NIL:

Ensure that both maps are 600px high.

### 8.1431.2 File

Defined in file src/terrain.lisp.

## 8.1432 Tootsville::⊕Post-Check-Map-Widths

### 8.1432.1 Function

⊕Post-Check-Map-Widths names a function, with lambda list NIL:

Ensure that both maps are 800px wide.

### 8.1432.2 File

Defined in file src/terrain.lisp.

## 8.1433 Tootsville::⊕Post-Ensure-Package-Imports-From-Oliphant-Are-Available

### 8.1433.1 Function

⊕Post-Ensure-Package-Imports-From-Oliphant-Are-Available names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Ensure-Package-Imports-From-Oliphant-Are-Available.

### 8.1433.2 File

Defined in file src/package-post.lisp.

## 8.1434 Tootsville::⊕Post-Extract-Plist-Path-1

### 8.1434.1 Function

⊕Post-Extract-Plist-Path-1 names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Extract-Plist-Path-1.

### 8.1434.2 File

Defined in file src/version.lisp.

## 8.1435 Tootsville::⊕Post-Extract-Plist-Path-2

### 8.1435.1 Function

⊕Post-Extract-Plist-Path-2 names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Extract-Plist-Path-2.

### 8.1435.2 File

Defined in file src/version.lisp.



## 8.1436 Tootsville::⊕Post-Extract-Plist-Path-3

### 8.1436.1 Function

⊕Post-Extract-Plist-Path-3 names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Extract-Plist-Path-3.

### 8.1436.2 File

Defined in file src/version.lisp.

## 8.1437 Tootsville::⊕Post-Extract-Plist-Path-4

### 8.1437.1 Function

⊕Post-Extract-Plist-Path-4 names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Extract-Plist-Path-4.

### 8.1437.2 File

Defined in file src/version.lisp.

## 8.1438 Tootsville::⊕Post-Good-Uri-Amazon-S3

### 8.1438.1 Function

⊕Post-Good-Uri-Amazon-S3 names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Good-Uri-Amazon-S3.

### 8.1438.2 File

Defined in file `src/types/uri-types.lisp`.

## 8.1439 Tootsville::⊕Post-Good-Uri-Tootsville.Org

### 8.1439.1 Function

⊕Post-Good-Uri-Tootsville.Org names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Good-Uri-Tootsville.Org.

### 8.1439.2 File

Defined in file `src/types/uri-types.lisp`.

## 8.1440 Tootsville::⊕Post-Good-Uri-With-Query-String

### 8.1440.1 Function

⊕Post-Good-Uri-With-Query-String names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Good-Uri-With-Query-String.

### 8.1440.2 File

Defined in file src/types/uri-types.lisp.

## 8.1441 Tootsville::⊕Post-Group-Plists

### 8.1441.1 Function

⊕Post-Group-Plists names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Group-Plists.

### 8.1441.2 File

Defined in file `src/endpoints/slash-meta-game.lisp`.

## 8.1442 Tootsville::⊕Post-Host-Name-Like-S3.Amazonaws.Com

### 8.1442.1 Function

⊕Post-Host-Name-Like-S3.Amazonaws.Com names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Host-Name-Like-S3.Amazonaws.Com.

### 8.1442.2 File

Defined in file src/types/uri-types.lisp.

## 8.1443 Tootsville::⊕Post-Host-Name-Like-Star-Hope.Org

### 8.1443.1 Function

⊕Post-Host-Name-Like-Star-Hope.Org names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Host-Name-Like-Star-Hope.Org.

### 8.1443.2 File

Defined in file src/types/uri-types.lisp.



## 8.1444 Tootsville::⊕Post-Host-Name-Like-Tootsville.Org

### 8.1444.1 Function

⊕Post-Host-Name-Like-Tootsville.Org names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Host-Name-Like-Tootsville.Org.

### 8.1444.2 File

Defined in file src/types/uri-types.lisp.

## **8.1445 Tootsville::⊕Post-Host-Name-Like-Www.Gov.Uk**

### **8.1445.1 Function**

⊕Post-Host-Name-Like-Www.Gov.Uk names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Host-Name-Like-Www.Gov.Uk.

### **8.1445.2 File**

Defined in file src/types/uri-types.lisp.

## 8.1446 Tootsville::⊕Post-Host-Name-Like- Www.Tootsville.Org

### 8.1446.1 Function

⊕Post-Host-Name-Like-Www.Tootsville.Org names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Host-Name-Like-Www.Tootsville.Org.

### 8.1446.2 File

Defined in file src/types/uri-types.lisp.

## **8.1447 Tootsville::⊕Post-Memcached-Quick-Test**

### **8.1447.1 Function**

⊕Post-Memcached-Quick-Test names a function, with lambda list NIL:

Quick test provided by CL-MemCacheD library

### **8.1447.2 File**

Defined in file src/db/memcached.lisp.

## 8.1448 Tootsville::⊕Post-Memcached-Random-Number-Test

### 8.1448.1 Function

⊕Post-Memcached-Random-Number-Test names a function, with lambda list NIL:

Store and fetch a random number

### 8.1448.2 File

Defined in file src/db/memcached.lisp.

## 8.1449 Tootsville::⊕Post-Normalize-Url-Collapse- / ./ -To- /

### 8.1449.1 Function

⊕Post-Normalize-Url-Collapse- / ./ -To- / names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Normalize-Url-Collapse- / ./ -To- /.

### 8.1449.2 File

Defined in file src/endpoints/gossip/alex/alex.lisp.

## 8.1450 Tootsville::⊕Post-Normalize-Url-Collapse- / / -To- /

### 8.1450.1 Function

⊕Post-Normalize-Url-Collapse- / / -To- / names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Normalize-Url-Collapse-// -To- /.

### 8.1450.2 File

Defined in file src/endpoints/gossip/alex/alex.lisp.

## 8.1451 Tootsville::⊕Post-Normalize-Url-Handle-../ -Chains

### 8.1451.1 Function

⊕Post-Normalize-Url-Handle-../ -Chains names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Normalize-Url-Handle-../-Chains.

### 8.1451.2 File

Defined in file src/endpoints/gossip/alex/alex.lisp.



## 8.1452 Tootsville::⊕Post-Normalize-Url-Hostname-Downcased

### 8.1452.1 Function

⊕Post-Normalize-Url-Hostname-Downcased names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Normalize-Url-Hostname-Downcased.

### 8.1452.2 File

Defined in file `src/endpoints/gossip/alex/alex.lisp`.

## 8.1453 Tootsville::⊕Post-Normalize-Url-Include-Unusual-Http-Port

### 8.1453.1 Function

⊕Post-Normalize-Url-Include-Unusual-Http-Port names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Normalize-Url-Include-Unusual-Http-Port.

### 8.1453.2 File

Defined in file `src/endpoints/gossip/alex/alex.lisp`.

## 8.1454 Tootsville::⊕Post-Normalize-Url-Include-Unusual-Https-Port

### 8.1454.1 Function

⊕Post-Normalize-Url-Include-Unusual-Https-Port names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Normalize-Url-Include-Unusual-Https-Port.

### 8.1454.2 File

Defined in file `src/endpoints/gossip/alex/alex.lisp`.

## 8.1455 Tootsville::⊕Post-Normalize-Url-Leave-%Xx-Encoded-Bytes

### 8.1455.1 Function

⊕Post-Normalize-Url-Leave-%Xx-Encoded-Bytes names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Normalize-Url-Leave-%Xx-Encoded-Bytes.

### 8.1455.2 File

Defined in file `src/endpoints/gossip/alex/alex.lisp`.

## 8.1456 Tootsville::⊕Post-Normalize-Url-Omit-Default-Https-Port

### 8.1456.1 Function

⊕Post-Normalize-Url-Omit-Default-Https-Port names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Normalize-Url-Omit-Default-Https-Port.

### 8.1456.2 File

Defined in file `src/endpoints/gossip/alex/alex.lisp`.

## 8.1457 Tootsville::⊕Post-Normalize-Url-Omit-Default-Http-Port

### 8.1457.1 Function

⊕Post-Normalize-Url-Omit-Default-Http-Port names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Normalize-Url-Omit-Default-Http-Port.

### 8.1457.2 File

Defined in file `src/endpoints/gossip/alex/alex.lisp`.

## 8.1458 Tootsville::⊕Post-Normalize-Url-Protocol-Downcased

### 8.1458.1 Function

⊕Post-Normalize-Url-Protocol-Downcased names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Normalize-Url-Protocol-Downcased.

### 8.1458.2 File

Defined in file `src/endpoints/gossip/alexa/alexa.lisp`.

## 8.1459 Tootsville::⊕Post-Normalize-Url-Treat-../ -As-Up

### 8.1459.1 Function

⊕Post-Normalize-Url-Treat-../ -As-Up names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Normalize-Url-Treat-../-As-Up.

### 8.1459.2 File

Defined in file src/endpoints/gossip/alex/alex.lisp.



## 8.1460 Tootsville::⊕Post-Normalize-Url-Un%Xx-Escape-Basic-Ascii

### 8.1460.1 Function

⊕Post-Normalize-Url-Un%Xx-Escape-Basic-Ascii names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Normalize-Url-Un%Xx-Escape-Basic-Ascii.

### 8.1460.2 File

Defined in file src/endpoints/gossip/alex/alex.lisp.

## 8.1461 Tootsville::⊕Post-Normalize-Url-Use-%20Not-+-For-Space

### 8.1461.1 Function

⊕Post-Normalize-Url-Use-%20Not-+-For-Space names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Normalize-Url-Use-%20Not-+-For-Space.

### 8.1461.2 File

Defined in file src/endpoints/gossip/alex/alex.lisp.

## 8.1462 Tootsville::⊕Post-Not-Host-Name-Like–Foo.Com

### 8.1462.1 Function

⊕Post-Not-Host-Name-Like–Foo.Com names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Not-Host-Name-Like–Foo.Com.

### 8.1462.2 File

Defined in file src/types/uri-types.lisp.

## **8.1463 Tootsville::⊕Post-Not-Host-Name-Like-10.0.0.10**

### **8.1463.1 Function**

⊕Post-Not-Host-Name-Like-10.0.0.10 names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Not-Host-Name-Like-10.0.0.10.

### **8.1463.2 File**

Defined in file src/types/uri-types.lisp.

## 8.1464 Tootsville::⊕Post-Not-Host-Name-Like-9foo.Com

### 8.1464.1 Function

⊕Post-Not-Host-Name-Like-9foo.Com names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Not-Host-Name-Like-9foo.Com.

### 8.1464.2 File

Defined in file src/types/uri-types.lisp.

## 8.1465 Tootsville::⊕Post-Not-Host-Name-Like-Bar.-Foo.Com

### 8.1465.1 Function

⊕Post-Not-Host-Name-Like-Bar.-Foo.Com names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Not-Host-Name-Like-Bar.-Foo.Com.

### 8.1465.2 File

Defined in file src/types/uri-types.lisp.

## 8.1466 Tootsville::⊕Post-Not-Host-Name-Like-Bar.9foo.Com

### 8.1466.1 Function

⊕Post-Not-Host-Name-Like-Bar.9foo.Com names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Not-Host-Name-Like-Bar.9foo.Com.

### 8.1466.2 File

Defined in file src/types/uri-types.lisp.

## 8.1467 Tootsville::⊕Post-Not-Host-Name-Like-Foo

### 8.1467.1 Function

⊕Post-Not-Host-Name-Like-Foo names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Not-Host-Name-Like-Foo.

### 8.1467.2 File

Defined in file `src/types/uri-types.lisp`.



## 8.1468 Tootsville::⊕Post-Not-Host-Name-Like-Foo-Foo.Com

### 8.1468.1 Function

⊕Post-Not-Host-Name-Like-Foo-Foo.Com names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Not-Host-Name-Like-Foo-Foo.Com.

### 8.1468.2 File

Defined in file src/types/uri-types.lisp.

## 8.1469 Tootsville::⊕Post-Not-Host-Name-Like-Foo-.Com

### 8.1469.1 Function

⊕Post-Not-Host-Name-Like-Foo-.Com names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Not-Host-Name-Like-Foo-.Com.

### 8.1469.2 File

Defined in file src/types/uri-types.lisp.

## 8.1470 Tootsville::⊕Post-Not-Host-Name-Like-Foo.12

### 8.1470.1 Function

⊕Post-Not-Host-Name-Like-Foo.12 names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Not-Host-Name-Like-Foo.12.

### 8.1470.2 File

Defined in file `src/types/uri-types.lisp`.

## **8.1471 Tootsville::⊕Post-Not-Host-Name-Like-Foo.X**

### **8.1471.1 Function**

⊕Post-Not-Host-Name-Like-Foo.X names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Not-Host-Name-Like-Foo.X.

### **8.1471.2 File**

Defined in file `src/types/uri-types.lisp`.

## 8.1472 Tootsville::⊕Post-Not-Host-Name-Like-.Ko

### 8.1472.1 Function

⊕Post-Not-Host-Name-Like-.Ko names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Not-Host-Name-Like-.Ko.

### 8.1472.2 File

Defined in file `src/types/uri-types.lisp`.

## 8.1473 Tootsville::⊕Post-Subheader-Field-Parses

### 8.1473.1 Function

⊕Post-Subheader-Field-Parses names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Subheader-Field-Parses.

### 8.1473.2 File

Defined in file src/auth/auth-firebase.lisp.

## 8.1474 Tootsville::⊕Post-Unit-Test-Flatten-Plist-Tree

### 8.1474.1 Function

⊕Post-Unit-Test-Flatten-Plist-Tree names a function, with lambda list NIL:

This is an undocumented Power-On Self-Test, named Unit-Test-Flatten-Plist-Tree.

### 8.1474.2 File

Defined in file src/http-error.lisp.





## 9 Package Twilio

## **9.1 Twilio::As-Response**

### **9.1.1 Macro**

As-Response names an undocumented macro, with lambda list (&BODY BODY).

### **9.1.2 File**

Defined in file `src/lib/twilio/twilio-simple.lisp`.

## 9.2 Twilio::Dial

### 9.2.1 Function

Dial names an undocumented function, with lambda list (DESTINATION-NUMBER &KEY ACTION ANSWER-ON-BRIDGE CALLER-ID HANGUP-ON-STAR METHOD RECORD RECORDING-STATUS-CALLBACK RECORDING-STATUS-CALLBACK-METHOD RECORDING-STATUS-CALLBACK-EVENT RING-TONE TIME-LIMIT TIMEOUT TRIM CLIENT CONFERENCE NUMBER QUEUE SIM SIP).

### 9.2.2 File

Defined in file `src/lib/twilio/twilio-simple.lisp`.

## 9.3 Twilio::Enqueue

### 9.3.1 Function

Enqueue names an undocumented function, with lambda list (&KEY ACTION METHOD WAIT-URL WAIT-URL-METHOD WORKFLOW-SID NAME TASK).

### 9.3.2 File

Defined in file `src/lib/twilio/twilio-simple.lisp`.

## 9.4 Twilio::Format-Language

### 9.4.1 Function

Format-Language names an undocumented function, with lambda list (SYMBOL).

### 9.4.2 File

Defined in file `src/lib/twilio/twilio-simple.lisp`.

## **9.5 Twilio::Hangup**

### **9.5.1 Function**

Hangup names an undocumented function, with lambda list NIL.

### **9.5.2 File**

Defined in file `src/lib/twilio/twilio-simple.lisp`.

## **9.6 Twilio::Leave**

### **9.6.1 Function**

Leave names an undocumented function, with lambda list NIL.

### **9.6.2 File**

Defined in file `src/lib/twilio/twilio-simple.lisp`.

## 9.7 Twilio::Message

### 9.7.1 Function

Message names an undocumented function, with lambda list (BODY-TEXT &KEY TO FROM ACTION METHOD MEDIA).

### 9.7.2 File

Defined in file `src/lib/twilio/twilio-simple.lisp`.



## 9.8 Twilio::Pause

### 9.8.1 Function

Pause names an undocumented function, with lambda list (&OPTIONAL (DURATION 1)).

### 9.8.2 File

Defined in file src/lib/twilio/twilio-simple.lisp.

## **9.9 Twilio::Play**

### **9.9.1 Function**

Play names an undocumented function, with lambda list (URI &KEY LOOP).

### **9.9.2 File**

Defined in file `src/lib/twilio/twilio-simple.lisp`.

## 9.10 Twilio::Play-Digits

### 9.10.1 Function

Play-Digits names an undocumented function, with lambda list (DIGITS &KEY LOOP).

### 9.10.2 File

Defined in file `src/lib/twilio/twilio-simple.lisp`.

## **9.11 Twilio::Record**

### **9.11.1 Function**

Record names an undocumented function, with lambda list (&REST \_).

### **9.11.2 File**

Defined in file `src/lib/twilio/twilio-simple.lisp`.

## 9.12 Twilio::Redirect

### 9.12.1 Function

Redirect names an undocumented function, with lambda list (URI &KEY METHOD).

### 9.12.2 File

Defined in file `src/lib/twilio/twilio-simple.lisp`.

## **9.13 Twilio::Reject**

### **9.13.1 Function**

Reject names an undocumented function, with lambda list (&KEY REASON).

### **9.13.2 File**

Defined in file `src/lib/twilio/twilio-simple.lisp`.

## 9.14 Twilio::Say

### 9.14.1 Function

Say names an undocumented function, with lambda list (TEXT &KEY VOICE LOOP LANGUAGE).

### 9.14.2 File

Defined in file `src/lib/twilio/twilio-simple.lisp`.

## 9.15 Twilio::With-Gather

### 9.15.1 Macro

With-Gather names an undocumented macro, with lambda list ((&REST KEYS &KEY ACTION (FINISH-ON-KEY) HINTS INPUT LANGUAGE METHOD NUM-DIGITS PARTIAL-RESULTS-CALLBACK PARTIAL-RESULTS-CALLBACK-METHOD (PROFANITY-FILTER) SPEECH-TIMEOUT TIMEOUT) &BODY BODY).

### 9.15.2 File

Defined in file src/lib/twilio/twilio-simple.lisp.



## 9.16 Twilio::With-Twilio-Params

### 9.16.1 Macro

With-Twilio-Params names an undocumented macro, with lambda list (NIL &BODY BODY).

### 9.16.2 File

Defined in file `src/lib/twilio/twilio-simple.lisp`.



## 10 Javascript

This chapter enumerates the client-side Javascript components of Tootsville.

This front-end is broken down into a number of modules, each of which is in its own namespace; often, a namespace is defined entirely by one source file, but not always.

## 10.1 Tootsville.AvatarBuilder.addNameTag

### 10.1.1 Function

Tootsville.AvatarBuilder.addNameTag is a function with lambda list: (avatar, model, scene)

Adds a nametag to an avatar. (Only in the main scene, for now.)

## 10.2 Tootsville.AvatarBuilder.build

### 10.2.1 Function

Tootsville.AvatarBuilder.build is a function with lambda list: (avatar, scene, finish)

Build an avatar based upon the description passed in.

The structure of the avatar description is as explained at Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534.

A duplicate of an existing avatar will not be created, but it may be updated.

## 10.3 Tootsville.AvatarBuilder.colorize

### 10.3.1 Function

Tootsville.AvatarBuilder.colorize is a function with lambda list: (avatar, model, scene, finish)

Colorize an Avatar and apply their pattern

## 10.4 Tootsville.AvatarBuilder.enableShadows

### 10.4.1 Function

Tootsville.AvatarBuilder.enableShadows is a function with lambda list: (object, scene)

Enable the object to cast shadows in the scene

## 10.5 Tootsville.AvatarBuilder.getPathForPattern

### 10.5.1 Function

Tootsville.AvatarBuilder.getPathForPattern is a function with lambda list: (pattern)



## 10.6 Tootsville.AvatarBuilder.loadAvatarBase

### 10.6.1 Function

Tootsville.AvatarBuilder.loadAvatarBase is a function with lambda list: (avatar, scene, finish)

Load the base avatar model from Jumbo.

## 10.7 Tootsville.AvatarBuilder.patterns

### 10.7.1 Variable

Patterns for Avatar Builder.

XXX Some day we might extract these from SVG files.

For now, these are manually extracted from the "d" attribute of an SVG shape.

These are used by the AvatarBuilder to draw patterns into the texture map on a Toot.

## 10.8 Tootsville.AvatarBuilder.postBuild

### 10.8.1 Function

Tootsville.AvatarBuilder.postBuild is a function with lambda list: (avatar, model, scene)

Actually build the avatar.

Don't call this directly, call Section 10.2 [Tootsville.AvatarBuilder.build], page 1783.

## 10.9 Tootsville.AvatarBuilder.rainbowColor

### 10.9.1 Function

Tootsville.AvatarBuilder.rainbowColor is a function with lambda list: (baseColor)

Pick a random color that does not match the avatar's skin

## 10.10 Tootsville.AvatarBuilder.rememberAvatar

### 10.10.1 Function

Tootsville.AvatarBuilder.rememberAvatar is a function with lambda list: (avatar, object, scene)

Add the avatar to the global list of avatars in the scene

## 10.11 Tootsville.AvatarBuilder.update

### 10.11.1 Function

Tootsville.AvatarBuilder.update is a function with lambda list: (avatar, model, scene, finish)

## 10.12 Tootsville.AvatarViewer.createCamera

### 10.12.1 Function

Tootsville.AvatarViewer.createCamera is a function with lambda list: (canvas, name)

Create a camera through which to observe the Avatar Viewer

## **10.13 Tootsville.AvatarViewer.createLight**

### **10.13.1 Function**

Tootsville.AvatarViewer.createLight is a function with lambda list: (canvas)

Create a light source for the AvatarViewer.



## **10.14 Tootsville.AvatarViewer.createScene**

### **10.14.1 Function**

Tootsville.AvatarViewer.createScene is a function with lambda list: (canvas)

Create a scene to contain the Avatar Viewer

## 10.15 Tootsville.AvatarViewer.createViewerInCanvas

### 10.15.1 Function

Tootsville.AvatarViewer.createViewerInCanvas is a function with lambda list: (toot, canvas, container)

Create a stand-alone Avatar Viewer in a CANVAS.

Create a 3D viewer with a single avatar in it, out of an arbitrary CANVAS element. Creates a Babylon 3D scene with just the avatar.

This is useful for paperdolls, character selection, &c.

The Toot info (avatar info) passed in must be in the form described at Section 8.1249 [TOOTSVILLE TOOT-INFO], page 1534, and will be ultimately passed to Section 10.2 [Tootsville.AvatarBuilder.build], page 1783, qv.

## 10.16 Tootsville.AvatarViewer.createViewerReally

### 10.16.1 Function

Tootsville.AvatarViewer.createViewerReally is a function with lambda list: (toot, canvas, container)

Render the AvatarViewer scene only once.

Then, grab a screenshot of it and put that into the canvas instead to free up the WebGL context.

## 10.17 Tootsville.AvatarViewer.getAvatar

### 10.17.1 Function

Tootsville.AvatarViewer.getAvatar is a function with lambda list: (character)

Get the avatar for the given character name.

Returns a promise which resolves into character information.

If character is falsey, returns a promise to an empty object.

## 10.18 Tootsville.FurnitureBuilder.build

### 10.18.1 Function

Tootsville.FurnitureBuilder.build is a function with lambda list: (item, scene, finish)

Given an item's description, load, colorize, and position its model.

Furniture items are described differently than avatars; the canonical description is at Section 8.770 [TOOTSVILLE ITEM-INFO], page 1055, *qv.*, and Section 8.787 [TOOTSVILLE ITEM-TEMPLATE-INFO], page 1072.

## 10.19 Tootsville.FurnitureBuilder.build2

### 10.19.1 Function

Tootsville.FurnitureBuilder.build2 is a function with lambda list: (item, model, scene, finish)

Finish construction of the object after it has been loaded by the asset manager.

## 10.20 Tootsville.FurnitureBuilder.colorize

### 10.20.1 Function

Tootsville.FurnitureBuilder.colorize is a function with lambda list: (item, model, scene, finish)

Colorize a furniture item

## 10.21 Tootsville.FurnitureBuilder.enableShadows

### 10.21.1 Function

Tootsville.FurnitureBuilder.enableShadows is a function with lambda list: (object, scene)

Enable the object to cast shadows in the scene



## 10.22 Tootsville.FurnitureBuilder.loadItemTemplate

### 10.22.1 Function

Tootsville.FurnitureBuilder.loadItemTemplate is a function with lambda list: (item, scene, finish)

Load an item template avatar from the assets server.

## 10.23 Tootsville.FurnitureBuilder.rememberItem

### 10.23.1 Function

Tootsville.FurnitureBuilder.rememberItem is a function with lambda list: (item, model, scene)

Stash a reference to the item in the scene.items object.

## 10.24 Tootsville.FurnitureBuilder.update

### 10.24.1 Function

Tootsville.FurnitureBuilder.update is a function with lambda list: (item, model, scene, finish)

WRITE ME . . . TODO update furniture . . .

## 10.25 Tootsville.Game.BallSystem.fastForward

### 10.25.1 Function

Tootsville.Game.BallSystem.fastForward is a function with lambda list:  $(\delta T)$

Simulate the passage of  $\Delta t$  time (in seconds)

## **10.26 Tootsville.Game.BallSystem.register**

### **10.26.1 Function**

Tootsville.Game.BallSystem.register is a function with lambda list: (entity, course)

## **10.27 Tootsville.Game.BallSystem.remove**

### **10.27.1 Function**

Tootsville.Game.BallSystem.remove is a function with lambda list: (entity)

## **10.28 Tootsville.Game.BallSystem.updateBalls**

### **10.28.1 Function**

Tootsville.Game.BallSystem.updateBalls is nullary function.

Update the position of all balls

## 10.29 Tootsville.Game.Commands.addFurniture

### 10.29.1 Function

Tootsville.Game.Commands.addFurniture is a function with lambda list: (d, u, r)

Alias for Section 10.70 [Tootsville.Game.Commands.setFurniture], page 1851, q.v.



## **10.30 Tootsville.Game.Commands.addToList**

### **10.30.1 Function**

Tootsville.Game.Commands.addToList is a function with lambda list: (d, u, r)

No longer used

## **10.31 Tootsville.Game.Commands.click**

### **10.31.1 Function**

Tootsville.Game.Commands.click is a function with lambda list: (d, u, r)

See Section 8.681 [TOOTSVILLE INFINITY-CLICK], page 939,

## 10.32 Tootsville.Game.Commands.createUserHouse

### 10.32.1 Function

Tootsville.Game.Commands.createUserHouse is a function with lambda list: (d, u, r)

Either claim the user's house and lot, or add a room to their house.

See Section 8.683 [TOOTSVILLE INFINITY-CREATE-USER-HOUSE], page 942,

## 10.33 Tootsville.Game.Commands.doff

### 10.33.1 Function

Tootsville.Game.Commands.doff is a function with lambda list: (d, u, r)

Doff an item

See Section 8.685 [TOOTSVILLE INFINITY-DOFF], page 944, and Section 8.686 [TOOTSVILLE INFINITY-DOFFF], page 945,

## 10.34 Tootsville.Game.Commands.don

### 10.34.1 Function

Tootsville.Game.Commands.don is a function with lambda list: (d, u, r)

Don an item

See Section 8.687 [TOOTSVILLE INFINITY-DON], page 946,

## 10.35 Tootsville.Game.Commands.echo

### 10.35.1 Function

Tootsville.Game.Commands.echo is a function with lambda list: (d, u, r)

Echoes back the supplied JSON (or ActionScript) object to the client.

This method exists solely for testing purposes.

See Section 8.688 [TOOTSVILLE INFINITY-ECHO], page 948,

## 10.36 Tootsville.Game.Commands.endEvent

### 10.36.1 Function

Tootsville.Game.Commands.endEvent is a function with lambda list: (d, u, r)

See Section 8.689 [TOOTSVILLE INFINITY-END-EVENT], page 949, (and Section 8.738 [TOOTSVILLE INFINITY-START-EVENT], page 1020)

## 10.37 Tootsville.Game.Commands.endevent

### 10.37.1 Function

Tootsville.Game.Commands.endevent is a function with lambda list: (d, u, r)

This method terminates an event (probably a minigame, but possibly a fountain) which was initiated by startEvent.

See Section 8.689 [TOOTSVILLE INFINITY-END-EVENT], page 949, (and Section 8.738 [TOOTSVILLE INFINITY-START-EVENT], page 1020, for context)



## 10.38 Tootsville.Game.Commands.finger

### 10.38.1 Function

Tootsville.Game.Commands.finger is a function with lambda list: (d, u, r)

Get public info for a list of (other) users.

See Section 8.691 [TOOTSVILLE INFINITY-FINGER], page 953,

## 10.39 Tootsville.Game.Commands.gameAction

### 10.39.1 Function

Tootsville.Game.Commands.gameAction is a function with lambda list: (d, u, r)

See Section 8.692 [TOOTSVILLE INFINITY-GAME-ACTION], page 954,

## 10.40 Tootsville.Game.Commands.getApple

### 10.40.1 Function

Tootsville.Game.Commands.getApple is a function with lambda list: (d, u, r)

Get the apple to get into, or out of, \$Eden

No longer needed

See Section 8.693 [TOOTSVILLE INFINITY-GET-APPLE], page 956,

## 10.41 Tootsville.Game.Commands.getAvatars

### 10.41.1 Function

Tootsville.Game.Commands.getAvatars is a function with lambda list: (d, u, r)

Get avatar data for a list of (other) users. cv. finger

See Section 8.694 [TOOTSVILLE INFINITY-GET-AVATARS], page 959, and Section 8.691 [TOOTSVILLE INFINITY-FINGER], page 953,

## 10.42 Tootsville.Game.Commands.getColorPalettes

### 10.42.1 Function

Tootsville.Game.Commands.getColorPalettes is a function with lambda list: (d, u, r)

See Section 8.695 [TOOTSVILLE INFINITY-GET-COLOR-PALETTES], page 960,

### 10.42.2 410 Gone

Removed.. This routine appeared to be unused by anyone in Romance 1.1 and was removed in 1.2.

### 10.42.3 Note

Not used in Tootsville any more. The analogous palettes in Li'l Vampies and Empires of the Air are being replaced with algorithmic checks, so this routine was removed in Appius 1.2.0.

## 10.43 Tootsville.Game.Commands.getInventory

### 10.43.1 Function

Tootsville.Game.Commands.getInventory is a function with lambda list: (d, u, r)  
get all inventory for an user — both active and inactive  
See Section 8.696 [TOOTSVILLE INFINITY-GET-INVENTORY], page 961,

## 10.44 Tootsville.Game.Commands.getInventoryByType

### 10.44.1 Function

Tootsville.Game.Commands.getInventoryByType is a function with lambda list: (d, u, r)

Get a subset of items from your own inventory

See Section 8.697 [TOOTSVILLE INFINITY-GET-INVENTORY-BY-TYPE], page 962,

## 10.45 Tootsville.Game.Commands.getOnlineUsers

### 10.45.1 Function

Tootsville.Game.Commands.getOnlineUsers is a function with lambda list: (d, u, r)

Get a list of users in a Zone, or in a Room.

This is an administrative function, only available to staff members.

See Section 8.699 [TOOTSVILLE INFINITY-GET-ONLINE-USERS], page 966,



## 10.46 Tootsville.Game.Commands.getRoomList

### 10.46.1 Function

Tootsville.Game.Commands.getRoomList is a function with lambda list: (d, u, r)

Get a list of all “well known” Rooms currently active/visible.

“Rooms” no longer exist. The “rooms” are now known as “planes.”

See Section 8.701 [TOOTSVILLE INFINITY-GET-ROOM-LIST], page 968,

## 10.47 Tootsville.Game.Commands.getServerTime

### 10.47.1 Function

Tootsville.Game.Commands.getServerTime is a function with lambda list: (d, u, r)

Send the server time to the client requesting it

Sends a JSON object with a single property, serverTime, with the current time in milliseconds (give or take transit time). This is the Unix time, not the Universal time.

See Section 8.703 [TOOTSVILLE INFINITY-GET-SERVER-TIME], page 973,

## 10.48 Tootsville.Game.Commands.getSessionApple

### 10.48.1 Function

Tootsville.Game.Commands.getSessionApple is a function with lambda list: (d, u, r)

Initialise a session key for stream or batch mode operations

Replies with { from: initSession, key: (OPAQUE-STRING) }

See Section 8.704 [TOOTSVILLE INFINITY-GET-SESSION-APPLE], page 974,

## 10.49 Tootsville.Game.Commands.getStoreItemInfo

### 10.49.1 Function

Tootsville.Game.Commands.getStoreItemInfo is a function with lambda list: (d, u, r)

See Section 8.705 [TOOTSVILLE INFINITY-GET-STORE-ITEM-INFO], page 975,

## 10.50 Tootsville.Game.Commands.getUserLists

### 10.50.1 Function

Tootsville.Game.Commands.getUserLists is a function with lambda list: (d, u, r)

Get the user's buddy list and ignore list.

See Section 8.706 [TOOTSVILLE INFINITY-GET-USER-LISTS], page 976,

## 10.51 Tootsville.Game.Commands.getWallet

### 10.51.1 Function

Tootsville.Game.Commands.getWallet is a function with lambda list: (d, u, r)

See Section 8.707 [TOOTSVILLE INFINITY-GET-WALLET], page 977,

## 10.52 Tootsville.Game.Commands.getZoneList

### 10.52.1 Function

Tootsville.Game.Commands.getZoneList is a function with lambda list: (d, u, r)

Get a list of all Zones currently active/visible.

See Section 8.708 [TOOTSVILLE INFINITY-GET-ZONE-LIST], page 978,

## 10.53 Tootsville.Game.Commands.give

### 10.53.1 Function

Tootsville.Game.Commands.give is a function with lambda list: (d, u, r)

Give an item to another user

See Section 8.709 [TOOTSVILLE INFINITY-GIVE], page 979,



## 10.54 Tootsville.Game.Commands.go

### 10.54.1 Function

Tootsville.Game.Commands.go is a function with lambda list: (d, u, r)

go to a place and/or perform a gesture

See Section 8.710 [TOOTSVILLE INFINITY-GO], page 980,

## 10.55 Tootsville.Game.Commands.initUserRoom

### 10.55.1 Function

Tootsville.Game.Commands.initUserRoom is a function with lambda list: (d, u, r)

Creates room named user/user's name/room ...

See Section 8.711 [TOOTSVILLE INFINITY-INIT-USER-ROOM], page 981,

## **10.56 Tootsville.Game.Commands.join**

### **10.56.1 Function**

Tootsville.Game.Commands.join is a function with lambda list: (d, u, r)

Join a room.

See Section 8.712 [TOOTSVILLE INFINITY-JOIN], page 982,

## 10.57 Tootsville.Game.Commands.login

### 10.57.1 Function

Tootsville.Game.Commands.login is a function with lambda list: (d, u, r)

Handle a login request

See Section 8.713 [TOOTSVILLE INFINITY-LOGIN], page 983,

## 10.58 Tootsville.Game.Commands.logout

### 10.58.1 Function

Tootsville.Game.Commands.logout is a function with lambda list: (d, u, r)

Log out of this game session (or zone)

See Section 8.714 [TOOTSVILLE INFINITY-LOGOUT], page 985,

## 10.59 Tootsville.Game.Commands.mailCustomerService

### 10.59.1 Function

Tootsville.Game.Commands.mailCustomerService is a function with lambda list: (d, u, r)  
send an eMail to customer service (feedback)

See Section 8.715 [TOOTSVILLE INFINITY-MAIL-CUSTOMER-SERVICE], page 986,

## 10.60 Tootsville.Game.Commands.peekAtInventory

### 10.60.1 Function

Tootsville.Game.Commands.peekAtInventory is a function with lambda list: (d, u, r)

Handle looking at other user's inventories

See Section 8.716 [TOOTSVILLE INFINITY-PEEK-AT-INVENTORY], page 987,

## 10.61 Tootsville.Game.Commands.ping

### 10.61.1 Function

Tootsville.Game.Commands.ping is a function with lambda list: (d, u, r)

Send a ping to the server to get back a pong.

See Section 8.717 [TOOTSVILLE INFINITY-PING], page 988,



## 10.62 Tootsville.Game.Commands.promptReply

### 10.62.1 Function

Tootsville.Game.Commands.promptReply is a function with lambda list: (d, u, r)

See Section 8.720 [TOOTSVILLE INFINITY-PROMPT-REPLY], page 991,

## 10.63 Tootsville.Game.Commands.removeFromList

### 10.63.1 Function

Tootsville.Game.Commands.removeFromList is a function with lambda list: (d, u, r)

Remove someone from a buddy list or ignore list.

See Section 8.723 [TOOTSVILLE INFINITY-REMOVE-FROM-LIST], page 996,

## 10.64 Tootsville.Game.Commands.reportBug

### 10.64.1 Function

Tootsville.Game.Commands.reportBug is a function with lambda list: (d, u, r)

This method allows the client to “phone home” to report a bug.

See Section 8.724 [TOOTSVILLE INFINITY-REPORT-BUG], page 997,

## 10.65 Tootsville.Game.Commands.reportUser

### 10.65.1 Function

Tootsville.Game.Commands.reportUser is a function with lambda list: (d, u, r)

Report an user to the moderator(s) on duty for breaking a rule

See Section 8.725 [TOOTSVILLE INFINITY-REPORT-USER], page 1002,

## 10.66 Tootsville.Game.Commands.requestBuddy

### 10.66.1 Function

Tootsville.Game.Commands.requestBuddy is a function with lambda list: (d, u, r)

Request adding a user to your buddy list (mutual-add) using the notification-based system

See Section 8.726 [TOOTSVILLE INFINITY-REQUEST-BUDDY], page 1003,

## 10.67 Tootsville.Game.Commands.sendOutOfBandMessage

### 10.67.1 Function

Tootsville.Game.Commands.sendOutOfBandMessage is a function with lambda list: (d, u, r)

Send an arbitrary JSON packet to another user, or all of the users in a room, out of the band of communications.

See Section 8.728 [TOOTSVILLE INFINITY-SEND-OUT-OF-BAND-MESSAGE], page 1006,

## 10.68 Tootsville.Game.Commands.serverTime

### 10.68.1 Function

Tootsville.Game.Commands.serverTime is a function with lambda list: (d, u, r)

This is used to synchronize universal time.

See Section 8.729 [TOOTSVILLE INFINITY-SERVER-TIME], page 1007,

## 10.69 Tootsville.Game.Commands.setAvatarColor

### 10.69.1 Function

Tootsville.Game.Commands.setAvatarColor is a function with lambda list: (d, u, r)

No longer used

See Section 8.730 [TOOTSVILLE INFINITY-SET-AVATAR-COLOR], page 1008,



## 10.70 Tootsville.Game.Commands.setFurniture

### 10.70.1 Function

Tootsville.Game.Commands.setFurniture is a function with lambda list: (d, u, r)

See Section 8.731 [TOOTSVILLE INFINITY-SET-FURNITURE], page 1009,

## 10.71 Tootsville.Game.Commands.spawnZone

### 10.71.1 Function

Tootsville.Game.Commands.spawnZone is a function with lambda list: (d, u, r)

No longer used

See Section 8.735 [TOOTSVILLE INFINITY-SPAWN-ZONE], page 1015,

## 10.72 Tootsville.Game.Commands.speak

### 10.72.1 Function

Tootsville.Game.Commands.speak is a function with lambda list: (d, u, r)  
speak

See Section 8.736 [TOOTSVILLE INFINITY-SPEAK], page 1016,

## 10.73 Tootsville.Game.Commands.startEvent

### 10.73.1 Function

Tootsville.Game.Commands.startEvent is a function with lambda list: (d, u, r)

See Section 8.738 [TOOTSVILLE INFINITY-START-EVENT], page 1020, (and Section 8.689 [TOOTSVILLE INFINITY-END-EVENT], page 949)

## 10.74 Tootsville.Game.Commands.useEquipment

### 10.74.1 Function

Tootsville.Game.Commands.useEquipment is a function with lambda list: (d, u, r)

See Section 8.741 [TOOTSVILLE INFINITY-USE-EQUIPMENT], page 1025,

## 10.75 Tootsville.Game.Commands.walk

### 10.75.1 Function

Tootsville.Game.Commands.walk is a function with lambda list: (d, u, r)

Begin walking along a straight path. Path specification:

The path is specified as a series of values joined by “~” (tilde) characters. Note that the coordinate triplets in particular are comma-delimited segments within a tilde-delimited segment.

- o            object (person) walking UUID
- sT           start time (Universal)
- eT           end time (Universal)
- sX, sY, sZ        start x, y, z
- eX, eY, eZ        end x, y, z

Example:

```
2A2FA700-FF51-11E8-A63B-94DE802CBFC4~3820353198~3820353298~0,0,0~0,100,0
```

This indicates that the Toot with UUID “2A2FA700-FF51-11E8-A63B-94DE802CBFC4” (which happens to be “Superstar”) began walking from 0,0,0 at Universal time 3820353198 (about 20:13 Eastern time on 22 January, 2021) and will finish walking in 100 seconds after that at 0,100,0.

cv: Section 8.744 [TOOTSVILLE INFINITY-WTL], page 1028, is the replacement command which uses a JSON equivalent for the “d” structure defined here.

### 10.75.2 Added in Romance 1.2

This replaced the “d” notation from Romance 1.1 with 1.2.0. In particular, the “z” coordinates are now required, and the sense of the “y” and “z” coordinates are different now.

### 10.75.3 Gossipnet only

This command is valid on the Gossipnet, but does not have a REST equivalent.

## 10.76 Tootsville.Game.Gatekeeper.admin

### 10.76.1 Function

Tootsville.Game.Gatekeeper.admin is a function with lambda list: (gram)

An administrative message

This message is presented by the Gossip Parrots in the UI.

**title**      The title of the message.

**message**    The body of the message.

**label**      The label of the message.

## 10.77 Tootsville.Game.Gatekeeper.avatars

### 10.77.1 Function

Tootsville.Game.Gatekeeper.avatars is a function with lambda list: (gram)

Receive a list of avatar info that describes an area of the world.

This is one observer's set of nearby avatars or objects.



## 10.78 Tootsville.Game.Gatekeeper.ayt

### 10.78.1 Function

Tootsville.Game.Gatekeeper.ayt is a function with lambda list: (gram)

Respond to AYT (Are You There) inquiry

If the server sees no activity for a long time, it'll send an Are You There (ayt) packet to verify that the client isn't just a zombie connection. We reply with a ping to show some activity.

## 10.79 Tootsville.Game.Gatekeeper.badgeUpdate

### 10.79.1 Function

Tootsville.Game.Gatekeeper.badgeUpdate is a function with lambda list: (gram)

We no longer have badges

## 10.80 Tootsville.Game.Gatekeeper.beam

### 10.80.1 Function

Tootsville.Game.Gatekeeper.beam is a function with lambda list: (gram)

The player has been teleported to the given coördinates and should now explore the surroundings (discover what objects are nearby, et al.).

The author must have the privilege to beam this player, or the signal should be discarded.

TODO UNIMPLEMENTED

## 10.81 Tootsville.Game.Gatekeeper.bots

### 10.81.1 Function

Tootsville.Game.Gatekeeper.bots is a function with lambda list: (gram)

No longer used.

## 10.82 Tootsville.Game.Gatekeeper.buddyList

### 10.82.1 Function

Tootsville.Game.Gatekeeper.buddyList is a function with lambda list: (gram)

WRITE ME — this function is not yet documented.

## 10.83 Tootsville.Game.Gatekeeper.buddyRequest

### 10.83.1 Function

Tootsville.Game.Gatekeeper.buddyRequest is a function with lambda list: (gram)

You have been requested to be someone's buddy.

See Section 8.726 [TOOTSVILLE INFINITY-REQUEST-BUDDY], page 1003,

### 10.83.2 Example

```
{ from: "buddyRequest",  
  status: true,  
  sender: "Pil",  
  signature: ""
```

## 10.84 Tootsville.Game.Gatekeeper.burgeon

### 10.84.1 Function

Tootsville.Game.Gatekeeper.burgeon is a function with lambda list: (gram)

Burgeon the Toot on logging back in.

TODO extended attributes are ignored

XXX d3 objects are ignored. `wt1` is the only supported course type for now.

## 10.85 Tootsville.Game.Gatekeeper.bye

### 10.85.1 Function

Tootsville.Game.Gatekeeper.bye is a function with lambda list: (gram)

The user has left the game.

The user named **n** with UUID **u** has left the game.

This destroys their avatar.



## **10.86 Tootsville.Game.Gatekeeper.c**

### **10.86.1 Function**

Tootsville.Game.Gatekeeper.c is a function with lambda list: (gram)

WRITEME — this function is not yet documented.

## 10.87 Tootsville.Game.Gatekeeper.earning

### 10.87.1 Function

Tootsville.Game.Gatekeeper.earning is a function with lambda list: (gram)

Player has received money (peanuts) or fairy dust.

TODO: update wallet displays with an animation.

## 10.88 Tootsville.Game.Gatekeeper.endEvent

### 10.88.1 Function

Tootsville.Game.Gatekeeper.endEvent is a function with lambda list: (gram)

End an event begun by startEvent.

Earn peanuts for event participation.

UNIMPLEMENTED. See also Section 10.87 [Tootsville.Game.Gatekeeper.earning], page 1868,

## 10.89 Tootsville.Game.Gatekeeper.forceMove

### 10.89.1 Function

Tootsville.Game.Gatekeeper.forceMove is a function with lambda list: (gram)

Force the character to move to (x,y,z) local.

**status**      Must be true

**x,y,z**      Local coördinates

## **10.90 Tootsville.Game.Gatekeeper.gameAction**

### **10.90.1 Function**

Tootsville.Game.Gatekeeper.gameAction is a function with lambda list: (gram)

Not currently in use. UNIMPLEMENTED.

## 10.91 Tootsville.Game.Gatekeeper.getApple

### 10.91.1 Function

Tootsville.Game.Gatekeeper.getApple is a function with lambda list: (gram)

Get an apple from the server for a child's login.

Upon receipt, create the SHA1-digest-hex password code and submit an Section 8.713 [TOOTSVILLE INFINITY-LOGIN], page 983, packet.

See Section 8.693 [TOOTSVILLE INFINITY-GET-APPLE], page 956, for an overview of the login process for children.

If status = false, retries to call `getApple` again.

## 10.92 Tootsville.Game.Gatekeeper.getAvailableHouses

### 10.92.1 Function

Tootsville.Game.Gatekeeper.getAvailableHouses is a function with lambda list: (gram)

Get a list of lots and house types which are available for starter houses to build on a lot.

**status** This packet is ignored unless **status** is **true**

**lots** The list of available lots in the given neighborhood. FIXME: format?

**houses** The list of available houses in the given neighborhood. FIXME: format?

**neighborhood**  
The selected neighborhood.

## **10.93 Tootsville.Game.Gatekeeper.getAwardRankings**

### **10.93.1 Function**

Tootsville.Game.Gatekeeper.getAwardRankings is a function with lambda list: (gram)

No longer used. Ignored.



## **10.94 Tootsville.Game.Gatekeeper.getColorPalettes**

### **10.94.1 Function**

Tootsville.Game.Gatekeeper.getColorPalettes is a function with lambda list: (gram)

No longer used.

## 10.95 Tootsville.Game.Gatekeeper.getMailInBox

### 10.95.1 Function

Tootsville.Game.Gatekeeper.getMailInBox is a function with lambda list: (gram)

Get an enumeration of messages in the player's SMS "inbox"

**status** If **true**, there are messages. If **false**, there are no messages and the **mail** object is not present.

**mail** An enumeration of message objects. Each message object has FIXME format.

UNIMPLEMENTED.

See Section 8.698 [TOOTSVILLE INFINITY-GET-MAIL-IN-BOX], page 964,

## 10.96 Tootsville.Game.Gatekeeper.getMailMessage

### 10.96.1 Function

Tootsville.Game.Gatekeeper.getMailMessage is a function with lambda list: (gram)

Fetch one SMS message by UUID.

**status** If `true`, a message was found.

**message** The message object. This contains the message uuid, sender, and body in FIXME format.

See Section 8.698 [TOOTSVILLE INFINITY-GET-MAIL-IN-BOX], page 964,

## **10.97 Tootsville.Game.Gatekeeper.getStoreItems**

### **10.97.1 Function**

Tootsville.Game.Gatekeeper.getStoreItems is a function with lambda list: (gram)

No longer used.

## 10.98 Tootsville.Game.Gatekeeper.getUserLists

### 10.98.1 Function

Tootsville.Game.Gatekeeper.getUserLists is a function with lambda list: (gram)

The user's buddy list and ignore list.

buddyList

ignoreList

## 10.99 Tootsville.Game.Gatekeeper.goToWeb

### 10.99.1 Function

Tootsville.Game.Gatekeeper.goToWeb is a function with lambda list: (gram)

This packet instructs the user to leave the game and go to a different web site.

The packet's `url` is immediately loaded, kicking them out of Tootsville.

## 10.100 Tootsville.Game.Gatekeeper.initUserRoom

### 10.100.1 Function

Tootsville.Game.Gatekeeper.initUserRoom is a function with lambda list: (gram)

Add a room (including a first room) to a user's house/lot.

UNIMPLEMENTED. it's unclear if this is actually needed on the client side regardless.

## 10.101 Tootsville.Game.Gatekeeper.inventory

### 10.101.1 Function

Tootsville.Game.Gatekeeper.inventory is a function with lambda list: (gram)

No longer handled by  $\infty$  mode protocols; now, fetched directly from the game server over REST API. FIXME not necessarily true



## 10.102 Tootsville.Game.Gatekeeper.joinOK

### 10.102.1 Function

Tootsville.Game.Gatekeeper.joinOK is a function with lambda list: (gram)

WRITEME — this function is not yet documented.

## 10.103 Tootsville.Game.Gatekeeper.kick

### 10.103.1 Function

Tootsville.Game.Gatekeeper.kick is a function with lambda list: (gram)

The user has been kicked out. Quit the game.

TODO kick reason display

## 10.104 Tootsville.Game.Gatekeeper.logOK

### 10.104.1 Function

Tootsville.Game.Gatekeeper.logOK is a function with lambda list: (gram)

Acknowledge a new player's login

neighbor: next-hop neighbor's UUID for peer-to-peer connections

Note that this message (only) uses `_cmd` as its attribute rather than `c` or `from` for historical reasons.

This message is usually unicast.

Hides Toots view, if it was present. Displays any `motd` on server connection.

UNIMPLEMENTED TODO peer connections.

## 10.105 Tootsville.Game.Gatekeeper.login

### 10.105.1 Function

Tootsville.Game.Gatekeeper.login is a function with lambda list: (gram)

Response to a login attempt (for a child).

On success, displays the `child-wait` overlay, awaiting parental approval.

On failure, the gossip parrots reveal the error message and detailed error code.

## 10.106 Tootsville.Game.Gatekeeper.migrate

### 10.106.1 Function

Tootsville.Game.Gatekeeper.migrate is a function with lambda list: (gram)

Migrate from the current websockets server to another one.

Optional parameter ‘newConnection’ can be used to provide a new ‘Section 10.490 [Tootsville.host], page 2271’ `game` URL. Normally, though, this will be `#same`, indicating no change — the load balancers should take care of things.

## 10.107 Tootsville.Game.Gatekeeper.newScript

### 10.107.1 Function

Tootsville.Game.Gatekeeper.newScript is a function with lambda list: (gram)

Load a new Javascript file pushed from the server.

This is potentially useful for patching the game on-the-fly.

The expected syntax is something like:

```
{ from: "newScript",  
  status: true,  
  script: "/path/to/script.js" }
```

See Section 7.63 [TOOTSVILLE-USER PUSH-SCRIPT], page 194, for one way to generate these packets.

## 10.108 Tootsville.Game.Gatekeeper.outOfBand

### 10.108.1 Function

Tootsville.Game.Gatekeeper.outOfBand is a function with lambda list: (gram)

General out-of-band messaging between users. Typically used for invitation to a location.

**type**        The type of out-of-band message  
**body**        The body of that message  
**status**      Must be **true** or the message will be ignored.

The contents of the **body** vary by **type**

### 10.108.2 Invitation

An invitation has type **invite**.

The body contains **FIXME**.

### 10.108.3 Response

**WRITEME**

### 10.108.4 To Room

**WRITEME**

## 10.109 Tootsville.Game.Gatekeeper.parentApproval

### 10.109.1 Function

Tootsville.Game.Gatekeeper.parentApproval is a function with lambda list: (gram)

Receive parent approval for a child's Toot to log in.

```
{ from: "parentApproval",  
status: true,  
until: 1611945540 }
```

```
{ from: "parentApproval",  
status: false }
```

If **status** is **true**, the parent has approved; if **status** is **false**, then the parent has denied permission.

The time until which the player is allowed to play is passed in **until** as a Unix time



## **10.110 Tootsville.Game.Gatekeeper.passport**

### **10.110.1 Function**

Tootsville.Game.Gatekeeper.passport is a function with lambda list: (gram)

Not currently used. UNIMPLEMENTED.

XXX bring back passports

## 10.111 Tootsville.Game.Gatekeeper.playWith

### 10.111.1 Function

Tootsville.Game.Gatekeeper.playWith is a function with lambda list: (gram)

The server has given us “permission” to play as the Toot we asked for — i.e. the second step of signing in is complete.

## **10.112 Tootsville.Game.Gatekeeper.postman**

### **10.112.1 Function**

Tootsville.Game.Gatekeeper.postman is a function with lambda list: (gram)  
Notification of new SMS message(s)

## 10.113 Tootsville.Game.Gatekeeper.prompt

### 10.113.1 Function

Tootsville.Game.Gatekeeper.prompt is a function with lambda list: (gram)

Display a server-pushed prompt and prepare to reply.

See Section 8.720 [TOOTSVILLE INFINITY-PROMPT-REPLY], page 991, for a discussion of the prompt system and the format of this datagram.

See Section 10.470 [Tootsville.UI.makePrompt], page 2251, for the implementation of the prompt dialog builder.

## 10.114 Tootsville.Game.Gatekeeper.pub

### 10.114.1 Function

Tootsville.Game.Gatekeeper.pub is a function with lambda list: (gram)

Public message (speech)

See Section 8.736 [TOOTSVILLE INFINITY-SPEAK], page 1016, for how speech is generated, or Section 8.1289 [TOOTSVILLE TOOT-SPEAK], page 1576.

The **pub** packet contains the speaker's name and UUID, the text spoken, and (optionally) whether it was whispered or shouted.

### 10.114.2 Example

```
{ from: "pub",  
  status: true,  
  u: "Toot-name",  
  t: "This is what I say",  
  x: "whisper",  
  id: "2259E5F5-CDED-4A6A-AE68-1C4BA481CB7C" }
```

## **10.115 Tootsville.Game.Gatekeeper.purchase**

### **10.115.1 Function**

Tootsville.Game.Gatekeeper.purchase is a function with lambda list: (gram)

Not currently used.

## 10.116 Tootsville.Game.Gatekeeper.quiesce

### 10.116.1 Function

Tootsville.Game.Gatekeeper.quiesce is a function with lambda list: (gram)

Respond to a request to quiesce.

See Section 8.721 [TOOTSVILLE INFINITY-QUIESCE], page 994, Section 10.184 [Tootsville.Game.Nav.quiesce], page 1965,

## **10.117 Tootsville.Game.Gatekeeper.reportBug**

### **10.117.1 Function**

Tootsville.Game.Gatekeeper.reportBug is a function with lambda list: (gram)

Acknowledgment of a bug report



## 10.118 Tootsville.Game.Gatekeeper.rv

### 10.118.1 Function

Tootsville.Game.Gatekeeper.rv is a function with lambda list: (gram)

React to “room variables”.

See Section 8.819 [TOOTSVILLE LOCAL-ROOM-VARS], page 1104, for a discussion. Room variables are a general communication channel of miscellaneous information about the game world. When received, they are interpreted to change or update information about the player’s surroundings.

Currently, the client supports the following room variable types:

`sky`, `weather`, `floor` (ignored), `item`, `itm2`, `furniture`, `text`, `zone` (place).

Destruction of objects is UNIMPLEMENTED in the client currently (TODO)

### 10.118.2 See Also

Section 8.702 [TOOTSVILLE INFINITY-GET-ROOM-VARS], page 969, Section 10.309 [Tootsville.SkyBuilder.buildMatchingSky], page 2090, Section 10.310 [Tootsville.SkyBuilder.buildMatchingWeather], page 2091, Section 10.303 [Tootsville.SceneBuilder.addItem1], page 2084, Section 10.304 [Tootsville.SceneBuilder.addItem2], page 2085, Section 10.302 [Tootsville.SceneBuilder.addFurn], page 2083, Section 10.306 [Tootsville.SceneBuilder.addText], page 2087, Section 10.305 [Tootsville.SceneBuilder.addPlace], page 2086,

## 10.119 Tootsville.Game.Gatekeeper.scoreUpdate

### 10.119.1 Function

Tootsville.Game.Gatekeeper.scoreUpdate is a function with lambda list: (gram)

Used to be used for minigame scores; not currently used.

UNIMPLEMENTED. Display the rank and score in an overlay.

## **10.120 Tootsville.Game.Gatekeeper.sendMessage**

### **10.120.1 Function**

Tootsville.Game.Gatekeeper.sendMessage is a function with lambda list: (gram)  
Confirmation that a message were sent.

## 10.121 Tootsville.Game.Gatekeeper.serverTime

### 10.121.1 Function

Tootsville.Game.Gatekeeper.serverTime is a function with lambda list: (gram)

Received acknowledgement of the server's time.

**status** should be true

**serverTime**

In milliseconds since Unix epoch

**gameTime** In milliseconds since Unix epoch

UNIMPLEMENTED. Should show the game timer, if present.

UNIMPLEMENTED. Should update estimated server lag?

## 10.122 Tootsville.Game.Gatekeeper.startEvent

### 10.122.1 Function

Tootsville.Game.Gatekeeper.startEvent is a function with lambda list: (gram)

Mostly just for fountains, now

UNIMPLEMENTED

## 10.123 Tootsville.Game.Gatekeeper.tootList

### 10.123.1 Function

Tootsville.Game.Gatekeeper.tootList is a function with lambda list: (gram)

Receive a list of Toots from the server.

When “status” is false, the user has no Toots.

## 10.124 Tootsville.Game.Gatekeeper.wardrobe

### 10.124.1 Function

Tootsville.Game.Gatekeeper.wardrobe is a function with lambda list: (gram)

Obtains visible information about one avatar at a time; not only clothing, but any equipped object.

## 10.125 Tootsville.Game.Gatekeeper.wtl

### 10.125.1 Function

Tootsville.Game.Gatekeeper.wtl is a function with lambda list: (gram)

Walk The Line

This is the method that Romance 2.0 uses for moving avatars in the game world.

See also: Section 8.744 [TOOTSVILLE INFINITY-WTL], page 1028,

There are some bookkeeping side-effects here:

- If the avatar named is not already known, we'll try to look them up with **finger**, but we'll also try to remember their WTL course.
- If the name and UUID for a character mismatch, we'll discard the packet.
- If the avatar is known, but has no associated model, we'll try to build the model.

These are basically part of the attempt to synchronize the client in the presence of missing information, which should not happen often.



## 10.126 Tootsville.Game.GravitySystem.fastForward

### 10.126.1 Function

Tootsville.Game.GravitySystem.fastForward is a function with lambda list:  $(\delta T)$

Simulate the passage of  $\Delta t$  time (in seconds)

## 10.127 Tootsville.Game.GravitySystem.register

### 10.127.1 Function

Tootsville.Game.GravitySystem.register is a function with lambda list: (entity)

Makes an entity's model subject to gravity.

## 10.128 Tootsville.Game.GravitySystem.updateEntityGravity

### 10.128.1 Function

Tootsville.Game.GravitySystem.updateEntityGravity is a function with lambda list: (entity)

## **10.129 Tootsville.Game.GravitySystem.updateGravity**

### **10.129.1 Function**

Tootsville.Game.GravitySystem.updateGravity is nullary function.

## **10.130 Tootsville.Game.GrowthSystem.evolve**

### **10.130.1 Function**

Tootsville.Game.GrowthSystem.evolve is a function with lambda list: (entity)

## 10.131 Tootsville.Game.GrowthSystem.fastForward

### 10.131.1 Function

Tootsville.Game.GrowthSystem.fastForward is a function with lambda list: ( $\delta T$ )

Simulate the passage of  $\Delta t$  time (in seconds)

## **10.132 Tootsville.Game.GrowthSystem.grow**

### **10.132.1 Function**

Tootsville.Game.GrowthSystem.grow is a function with lambda list: (entity)

## 10.133 Tootsville.Game.GrowthSystem.register

### 10.133.1 Function

Tootsville.Game.GrowthSystem.register is a function with lambda list: (entity)

Register an entity that can grow.



## **10.134 Tootsville.Game.GrowthSystem.remove**

### **10.134.1 Function**

Tootsville.Game.GrowthSystem.remove is a function with lambda list: (entity)

## **10.135 Tootsville.Game.GrowthSystem.updateGrowth**

### **10.135.1 Function**

Tootsville.Game.GrowthSystem.updateGrowth is nullary function.

Update the growth of all eligible entities

## 10.136 Tootsville.Game.MissileSystem.fastForward

### 10.136.1 Function

Tootsville.Game.MissileSystem.fastForward is a function with lambda list: ( $\delta T$ )

Simulate the passage of  $\Delta t$  time (in seconds)

## **10.137 Tootsville.Game.MissileSystem.register**

### **10.137.1 Function**

Tootsville.Game.MissileSystem.register is a function with lambda list: (entity, course)

## **10.138 Tootsville.Game.MissileSystem.remove**

### **10.138.1 Function**

Tootsville.Game.MissileSystem.remove is a function with lambda list: (entity)

## **10.139 Tootsville.Game.MissileSystem.updateMissiles**

### **10.139.1 Function**

Tootsville.Game.MissileSystem.updateMissiles is nullary function.

Update the position of all missiles

## 10.140 Tootsville.Game.NPC.Collector.fastForward

### 10.140.1 Function

Tootsville.Game.NPC.Collector.fastForward is a function with lambda list: (npc)

Fast-forward the NPC for  $\Delta t$  seconds.

## **10.141 Tootsville.Game.NPC.Collector.register**

### **10.141.1 Function**

Tootsville.Game.NPC.Collector.register is a function with lambda list: (npc)

Register an NPC as a Collector



## 10.142 Tootsville.Game.NPC.Collector.updateNPC

### 10.142.1 Function

Tootsville.Game.NPC.Collector.updateNPC is a function with lambda list: (npc)

Update an NPC on the 50Hz game clock, if necessary.

## 10.143 Tootsville.Game.NPC.Cook.fastForward

### 10.143.1 Function

Tootsville.Game.NPC.Cook.fastForward is a function with lambda list: (npc)

Fast-forward the NPC for  $\Delta t$  seconds.

## **10.144 Tootsville.Game.NPC.Cook.register**

### **10.144.1 Function**

Tootsville.Game.NPC.Cook.register is a function with lambda list: (npc)

Register an NPC as a Cook

## **10.145 Tootsville.Game.NPC.Cook.updateNPC**

### **10.145.1 Function**

Tootsville.Game.NPC.Cook.updateNPC is a function with lambda list: (npc)

Update an NPC on the 50Hz game clock, if necessary.

## 10.146 Tootsville.Game.NPC.CroquetPlayer.fastForward

### 10.146.1 Function

Tootsville.Game.NPC.CroquetPlayer.fastForward is a function with lambda list: (npc)

Fast-forward the NPC for  $\Delta t$  seconds.

## **10.147 Tootsville.Game.NPC.CroquetPlayer.register**

### **10.147.1 Function**

Tootsville.Game.NPC.CroquetPlayer.register is a function with lambda list: (npc)

Register an NPC as a Croquet Player

## 10.148 Tootsville.Game.NPC.CroquetPlayer.updateNPC

### 10.148.1 Function

Tootsville.Game.NPC.CroquetPlayer.updateNPC is a function with lambda list: (npc)

Update an NPC on the 50Hz game clock, if necessary.

## 10.149 Tootsville.Game.NPC.Doodle.fastForward

### 10.149.1 Function

Tootsville.Game.NPC.Doodle.fastForward is a function with lambda list: (npc)

Fast-forward the NPC for  $\Delta t$  seconds.



## **10.150 Tootsville.Game.NPC.Doodle.register**

### **10.150.1 Function**

Tootsville.Game.NPC.Doodle.register is a function with lambda list: (npc)

Register an NPC as Doodle

## **10.151 Tootsville.Game.NPC.Doodle.updateNPC**

### **10.151.1 Function**

Tootsville.Game.NPC.Doodle.updateNPC is a function with lambda list: (npc)

Update an NPC on the 50Hz game clock, if necessary.

## 10.152 Tootsville.Game.NPC.Fetcher.fastForward

### 10.152.1 Function

Tootsville.Game.NPC.Fetcher.fastForward is a function with lambda list: (npc)

Fast-forward the NPC for  $\Delta t$  seconds.

## **10.153 Tootsville.Game.NPC.Fetcher.register**

### **10.153.1 Function**

Tootsville.Game.NPC.Fetcher.register is a function with lambda list: (npc)

Register an NPC as a Fetcher

## 10.154 Tootsville.Game.NPC.Fetcher.updateNPC

### 10.154.1 Function

Tootsville.Game.NPC.Fetcher.updateNPC is a function with lambda list: (npc)

Update an NPC on the 50Hz game clock, if necessary.

## 10.155 Tootsville.Game.NPC.JobWorker.fastForward

### 10.155.1 Function

Tootsville.Game.NPC.JobWorker.fastForward is a function with lambda list: (npc)

Fast-forward the NPC for  $\Delta t$  seconds.

## **10.156 Tootsville.Game.NPC.JobWorker.register**

### **10.156.1 Function**

Tootsville.Game.NPC.JobWorker.register is a function with lambda list: (npc)

Register an NPC as a Job Worker

## **10.157 Tootsville.Game.NPC.JobWorker.updateNPC**

### **10.157.1 Function**

Tootsville.Game.NPC.JobWorker.updateNPC is a function with lambda list: (npc)

Update an NPC on the 50Hz game clock, if necessary.



## 10.158 Tootsville.Game.NPC.MazeBuilder.fastForward

### 10.158.1 Function

Tootsville.Game.NPC.MazeBuilder.fastForward is a function with lambda list: (npc)

Fast-forward the NPC for  $\Delta t$  seconds.

## **10.159 Tootsville.Game.NPC.MazeBuilder.register**

### **10.159.1 Function**

Tootsville.Game.NPC.MazeBuilder.register is a function with lambda list: (npc)

Register an NPC as a Maze Builder

## **10.160 Tootsville.Game.NPC.MazeBuilder.updateNPC**

### **10.160.1 Function**

Tootsville.Game.NPC.MazeBuilder.updateNPC is a function with lambda list: (npc)

Update an NPC on the 50Hz game clock, if necessary.

## 10.161 Tootsville.Game.NPC.Sleeper.fastForward

### 10.161.1 Function

Tootsville.Game.NPC.Sleeper.fastForward is a function with lambda list: (npc)

Fast-forward the NPC for  $\Delta t$  seconds.

## 10.162 Tootsville.Game.NPC.Sleeper.register

### 10.162.1 Function

Tootsville.Game.NPC.Sleeper.register is a function with lambda list: (npc)

Register an NPC as a Sleeper.

Attributes: The time range during which the character wants to sleep. For most characters

## **10.163 Tootsville.Game.NPC.Sleeper.updateNPC**

### **10.163.1 Function**

Tootsville.Game.NPC.Sleeper.updateNPC is a function with lambda list: (npc)

Update an NPC on the 50Hz game clock, if necessary.

## 10.164 Tootsville.Game.NPC.TrolleyDriver.fastForward

### 10.164.1 Function

Tootsville.Game.NPC.TrolleyDriver.fastForward is a function with lambda list: (npc)

Fast-forward the NPC for  $\Delta t$  seconds.

## **10.165 Tootsville.Game.NPC.TrolleyDriver.register**

### **10.165.1 Function**

Tootsville.Game.NPC.TrolleyDriver.register is a function with lambda list: (npc)

Register an NPC as a Trolley Driver



## **10.166 Tootsville.Game.NPC.TrolleyDriver.updateNPC**

### **10.166.1 Function**

Tootsville.Game.NPC.TrolleyDriver.updateNPC is a function with lambda list: (npc)

Update an NPC on the 50Hz game clock, if necessary.

## 10.167 Tootsville.Game.NPC.Waiter.fastForward

### 10.167.1 Function

Tootsville.Game.NPC.Waiter.fastForward is a function with lambda list: (npc)

Fast-forward the NPC for  $\Delta t$  seconds.

## **10.168 Tootsville.Game.NPC.Waiter.register**

### **10.168.1 Function**

Tootsville.Game.NPC.Waiter.register is a function with lambda list: (npc)

Register an NPC as a Waiter

## **10.169 Tootsville.Game.NPC.Waiter.updateNPC**

### **10.169.1 Function**

Tootsville.Game.NPC.Waiter.updateNPC is a function with lambda list: (npc)

Update an NPC on the 50Hz game clock, if necessary.

## **10.170 Tootsville.Game.NPCSystem.burgeonNPC**

### **10.170.1 Function**

Tootsville.Game.NPCSystem.burgeonNPC is a function with lambda list: (npc)

## 10.171 Tootsville.Game.NPCSystem.fastForward

### 10.171.1 Function

Tootsville.Game.NPCSystem.fastForward is a function with lambda list: ( $\delta T$ )

Simulate the passage of  $\Delta t$  time (in seconds)

## 10.172 Tootsville.Game.NPCSystem.initNPCs

### 10.172.1 Function

Tootsville.Game.NPCSystem.initNPCs is nullary function.

NPCSystem initializer.

### 10.172.2 NPC System Overview

The NPC system operates in one of two modes, burgeoning and operating. When burgeoning, NPC data is fetched from the server and NPCs are fast-forwarded to the present time. When operating, each NPC should behave deterministically and the simulation should be equivalent on every observer's system, however, one system will act as the master and others will merely cross-check their observations.

## **10.173 Tootsville.Game.NPCSystem.nextBehavior**

### **10.173.1 Function**

Tootsville.Game.NPCSystem.nextBehavior is a function with lambda list: (npc)



## **10.174 Tootsville.Game.NPCSystem.register**

### **10.174.1 Function**

Tootsville.Game.NPCSystem.register is a function with lambda list: (avatar)

Register an NPC

## **10.175 Tootsville.Game.NPCSystem.updateNPC**

### **10.175.1 Function**

Tootsville.Game.NPCSystem.updateNPC is a function with lambda list: (npc)

## **10.176 Tootsville.Game.NPCSystem.updateNPCs**

### **10.176.1 Function**

Tootsville.Game.NPCSystem.updateNPCs is nullary function.

**10.177 Tootsville.Game.Nav.CAMERA\_MOVE\_SPEED****10.177.1 Variable**

The speed at which the camera moves

## **10.178 Tootsville.Game.Nav.RUN\_SPEED**

### **10.178.1 Variable**

The run speed of a Toot. Currently a constant.

**10.179 Tootsville.Game.Nav.WALK\_SPEED****10.179.1 Variable**

The walk speed of a Toot. Currently a constant.

## **10.180 Tootsville.Game.Nav.buildWTL**

### **10.180.1 Function**

Tootsville.Game.Nav.buildWTL is nullary function.

## **10.181 Tootsville.Game.Nav.collisionP**

### **10.181.1 Function**

Tootsville.Game.Nav.collisionP is a function with lambda list: (model, start, end)



## **10.182 Tootsville.Game.Nav.mergeObjects**

### **10.182.1 Function**

Tootsville.Game.Nav.mergeObjects is a function with lambda list: (into, from)

Merge keys of an object safely

## 10.183 Tootsville.Game.Nav.moveEntityOnCourse

### 10.183.1 Function

Tootsville.Game.Nav.moveEntityOnCourse is a function with lambda list: (entity, course)

Move an entity along a course, until its movement is interrupted by colliding with something else.

returns true when the course has been completed

## **10.184 Tootsville.Game.Nav.quiesce**

### **10.184.1 Function**

Tootsville.Game.Nav.quiesce is nullary function.

## **10.185 Tootsville.Game.Nav.runTo**

### **10.185.1 Function**

Tootsville.Game.Nav.runTo is a function with lambda list: (avatar, destinationPoint)

## **10.186 Tootsville.Game.Nav.sendWTL**

### **10.186.1 Function**

Tootsville.Game.Nav.sendWTL is nullary function.

## **10.187 Tootsville.Game.Nav.updateAvatar**

### **10.187.1 Function**

Tootsville.Game.Nav.updateAvatar is a function with lambda list: (avatar)

Update avatar's rotation & position.

## **10.188 Tootsville.Game.Nav.updateAvatars**

### **10.188.1 Function**

Tootsville.Game.Nav.updateAvatars is nullary function.

Update the position & rotation of every avatar

## **10.189 Tootsville.Game.Nav.updateCamera**

### **10.189.1 Function**

Tootsville.Game.Nav.updateCamera is nullary function.

Update the camera's position



## 10.190 Tootsville.Game.Nav.updateCameraDolly

### 10.190.1 Function

Tootsville.Game.Nav.updateCameraDolly is a function with lambda list: (model, cameraPosition)

Update the camera's dolly position (forward/back)

## **10.191 Tootsville.Game.Nav.updateCameraTruck**

### **10.191.1 Function**

Tootsville.Game.Nav.updateCameraTruck is a function with lambda list: (model, cameraPosition)

Update the camera's truck position (left/right)

## **10.192 Tootsville.Game.Nav.updateFacing**

### **10.192.1 Function**

Tootsville.Game.Nav.updateFacing is a function with lambda list: (avatar)

Update the avatar's facing direction to match desired direction.

## 10.193 Tootsville.Game.Nav.walkTheLine

### 10.193.1 Function

Tootsville.Game.Nav.walkTheLine is a function with lambda list: (avatar, destinationPoint, speed)

Set the course for the given avatar to lead toward the given destinationPoint.

TODO Allow directing a vehicle when mounted as its driver.

TODO Restrict movement when riding a vehicle.

## 10.194 Tootsville.Game.Speech.createBalloon

### 10.194.1 Function

Tootsville.Game.Speech.createBalloon is a function with lambda list: (words, extraClass)

Create a speech balloon containing “words” with CSS class “extraClass”.

## 10.195 Tootsville.Game.Speech.dispatchCommand

### 10.195.1 Function

Tootsville.Game.Speech.dispatchCommand is a function with lambda list: (commandLine)

Dispatch local ~ commands.

- ~ua        Displays the user agent information.
- ~lag       Provides the user's estimated lag in msec.
- ~ping     Send an Section 8.717 [TOOTSVILLE INFINITY-PING], page 988, message to the server.
- ~d20      Simulate rolling a d20 and speak aloud the results.
- ~credits   Display the client's credits.
- ~version   Display the version of the client

## **10.196 Tootsville.Game.Speech.removeSpeech**

### **10.196.1 Function**

Tootsville.Game.Speech.removeSpeech is a function with lambda list: (balloon)

The time has passed; remove a speech balloon.

## 10.197 Tootsville.Game.Speech.say

### 10.197.1 Function

Tootsville.Game.Speech.say is a function with lambda list: (words, extraClass, speaker)

Someone (maybe us) has spoken, so put up a speech balloon and play wawa.

If the speaker was not known already, finger the user.

Builds wawa with Section 10.456 [Tootsville.UI.WaWa.build], page 2237.



## **10.198 Tootsville.Game.Speech.updateSpeech**

### **10.198.1 Function**

Tootsville.Game.Speech.updateSpeech is nullary function.

Update speech balloons, expiring any that have aged out.

## 10.199 Tootsville.Game.Tools.axe

### 10.199.1 Function

Tootsville.Game.Tools.axe is a function with lambda list: (x, y, z)

Use an axe at the given coördinates.

Cut a branch or tree trunk.

## 10.200 Tootsville.Game.Tools.butterflyNet

### 10.200.1 Function

Tootsville.Game.Tools.butterflyNet is a function with lambda list: (x, y, z)

Use a butterfly net at the given coordinates.

Attempt to capture a bug.

## 10.201 Tootsville.Game.Tools.fishingRod

### 10.201.1 Function

Tootsville.Game.Tools.fishingRod is a function with lambda list: (x, y, z)

Use a fishing rod at the given coördinates.

Attempt to catch a fish.

## 10.202 Tootsville.Game.Tools.pickaxe

### 10.202.1 Function

Tootsville.Game.Tools.pickaxe is a function with lambda list: (x, y, z)

Use a pickaxe at the given coördinates.

Break apart a rock into smaller stones.

## 10.203 Tootsville.Game.Tools.sewingKit

### 10.203.1 Function

Tootsville.Game.Tools.sewingKit is a function with lambda list: (x, y, z)

Use a sewing kit at the given coördinates.

Used to create clothing from cloth.

## 10.204 Tootsville.Game.Tools.shovel

### 10.204.1 Function

Tootsville.Game.Tools.shovel is a function with lambda list: (x, y, z)

Use a shovel at the given coordinates.

A shovel creates a hole, and may reveal any hidden object buried there.

Using a shovel can also fill a hole, hiding any objects in the hole.

## 10.205 Tootsville.Game.Tools.wrench

### 10.205.1 Function

Tootsville.Game.Tools.wrench is a function with lambda list: (x, y, z)

Use a wrench at the given coördinates.

Given some building materials, combine them into a constructed object.



## 10.206 Tootsville.Game.Wardrobe

### 10.206.1 Variable

Clothing valences and conflicts, encoded for Javascript form. Compare to Section 8.1334 [TOOTSVILLE WEAR-SLOT], page 1621, which should be the canonical representation and used to populate this.

## **10.207 Tootsville.Game.Wardrobe.doff**

### **10.207.1 Function**

Tootsville.Game.Wardrobe.doff is a function with lambda list: (item)

## **10.208 Tootsville.Game.Wardrobe.don**

### **10.208.1 Function**

Tootsville.Game.Wardrobe.don is a function with lambda list: (item, slot)

Don an article of clothing on a wear slot.

## **10.209 Tootsville.Game.Wardrobe.drop**

### **10.209.1 Function**

Tootsville.Game.Wardrobe.drop is a function with lambda list: (item)

## 10.210 Tootsville.Game.Wardrobe.finalizeExchange

### 10.210.1 Function

Tootsville.Game.Wardrobe.finalizeExchange is a function with lambda list: (exchangePacket)

## **10.211 Tootsville.Game.Wardrobe.findBaseSlot**

### **10.211.1 Function**

Tootsville.Game.Wardrobe.findBaseSlot is a function with lambda list: (slot)

Find a base slot in the 3D model for clothing to mount.

## **10.212 Tootsville.Game.Wardrobe.inventory**

### **10.212.1 Function**

Tootsville.Game.Wardrobe.inventory is nullary function.

## **10.213 Tootsville.Game.Wardrobe.inventoryByKind**

### **10.213.1 Function**

Tootsville.Game.Wardrobe.inventoryByKind is a function with lambda list: (kind)



## 10.214 Tootsville.Game.Wardrobe.proposeExchange

### 10.214.1 Function

Tootsville.Game.Wardrobe.proposeExchange is a function with lambda list: (tradePartner, offerItems, demandItems)

Propose to exchange an item with a partner.

Offer `offerItems` to `tradePartner`, in return for `demandItems`.

## **10.215 Tootsville.Game.Wardrobe.readied**

### **10.215.1 Function**

Tootsville.Game.Wardrobe.readied is nullary function.

## **10.216 Tootsville.Game.Wardrobe.readiedP**

### **10.216.1 Function**

Tootsville.Game.Wardrobe.readiedP is a function with lambda list: (item)

## **10.217 Tootsville.Game.Wardrobe.ready**

### **10.217.1 Function**

Tootsville.Game.Wardrobe.ready is a function with lambda list: (item)

## **10.218 Tootsville.Game.Wardrobe.refresh**

### **10.218.1 Function**

Tootsville.Game.Wardrobe.refresh is nullary function.

## **10.219 Tootsville.Game.Wardrobe.signExchange**

### **10.219.1 Function**

Tootsville.Game.Wardrobe.signExchange is a function with lambda list: (exchangePacket)

Sign an exchange packet

## **10.220 Tootsville.Game.Wardrobe.take**

### **10.220.1 Function**

Tootsville.Game.Wardrobe.take is a function with lambda list: (item)

## **10.221 Tootsville.Game.Wardrobe.wearing**

### **10.221.1 Function**

Tootsville.Game.Wardrobe.wearing is nullary function.



## **10.222 Tootsville.Game.Wardrobe.wearingP**

### **10.222.1 Function**

Tootsville.Game.Wardrobe.wearingP is a function with lambda list: (item)

## **10.223 Tootsville.Game.bootstrap**

### **10.223.1 Function**

Tootsville.Game.bootstrap is nullary function.

Start up the game systems, end the loader phase, and start up the login process. This is the main "entry point" for the game.

## 10.224 Tootsville.Game.clickedOnItem

### 10.224.1 Function

Tootsville.Game.clickedOnItem is a function with lambda list: (itemNameString, picked-Event)

Respond to a click on an item (furniture)

## **10.225 Tootsville.Game.credits**

### **10.225.1 Function**

Tootsville.Game.credits is nullary function.

The credits for the game as a string.

Suitable for an “about this game” dialog box, credits crawl, or other presentation. This is expected to run long, over time, so take some efforts to ensure that there is scrolling or other affordance in place to accomodate a very long message.

## **10.226 Tootsville.Game.fastForward**

### **10.226.1 Function**

Tootsville.Game.fastForward is a function with lambda list: (sinceTime)

When burgeoning a region, fast-forward system effects to the present.

## **10.227 Tootsville.Game.hideWhenGameReady**

### **10.227.1 Function**

Tootsville.Game.hideWhenGameReady is nullary function.

Hide the loading overlay once the game is ready.

Call this function when the game is ready.

## 10.228 Tootsville.Game.interestingPoint

### 10.228.1 Function

Tootsville.Game.interestingPoint is a function with lambda list: (point)

Is the 'point' near to the center of current activity Tootsville.activity to be of interest to us? If the point is too far away, we may not care about it.

## **10.229 Tootsville.Game.lag**

### **10.229.1 Variable**

How much lag are we accommodating?



## **10.230 Tootsville.Game.stopSlowLoadingWatchdogs**

### **10.230.1 Function**

Tootsville.Game.stopSlowLoadingWatchdogs is nullary function.

Stop the watchdogs for slow loading.

Call this function once the loading has been completed enough.

## **10.231 Tootsville.Game.update**

### **10.231.1 Function**

Tootsville.Game.update is nullary function.

Update everything that operates on the 50Hz Game Tick clock.

## 10.232 Tootsville.Gossip.Parrot.ask

### 10.232.1 Function

Tootsville.Gossip.Parrot.ask is a function with lambda list: (title, message, replies)

Ask a question from the Gossip Parrot with multiple replies possible.

TODO: Document the format in which replies are submitted to this function.

## **10.233 Tootsville.Gossip.Parrot.done**

### **10.233.1 Function**

Tootsville.Gossip.Parrot.done is nullary function.

## **10.234 Tootsville.Gossip.Parrot.parrotErrorText**

### **10.234.1 Function**

Tootsville.Gossip.Parrot.parrotErrorText is a function with lambda list: (body)

Get the text which the Gossip Parrot should use to present an HTTP error.

TODO: document the input format.

## **10.235 Tootsville.Gossip.Parrot.say**

### **10.235.1 Function**

Tootsville.Gossip.Parrot.say is a function with lambda list: (title, message)

Give a message from the Gossip Parrot with an OK button.

## **10.236 Tootsville.Gossip.Parrot.show**

### **10.236.1 Function**

Tootsville.Gossip.Parrot.show is a function with lambda list: (reallyp)

Show or hide the parrot (based on reallyp flag).

## 10.237 Tootsville.Gossip.Parrot.ynP

### 10.237.1 Function

Tootsville.Gossip.Parrot.ynP is a function with lambda list: (title, message)

Ask a yes-or-no question from the Gossip Parrot.



## **10.238 Tootsville.Gossip.acceptOffer**

### **10.238.1 Function**

Tootsville.Gossip.acceptOffer is a function with lambda list: (offer)

Accept an offer which was exchanged

## 10.239 Tootsville.Gossip.closeInfinityMode

### 10.239.1 Function

Tootsville.Gossip.closeInfinityMode is a function with lambda list: (peer, event)

Remove a gossip PEER connection

## **10.240 Tootsville.Gossip.closeStreams**

### **10.240.1 Function**

Tootsville.Gossip.closeStreams is nullary function.

## 10.241 Tootsville.Gossip.connect

### 10.241.1 Function

Tootsville.Gossip.connect is a function with lambda list: (success)

Connect to the global gossip network.

Calls next function **success** on success.

## **10.242 Tootsville.Gossip.connectedP**

### **10.242.1 Function**

Tootsville.Gossip.connectedP is nullary function.

Are we connected to the global gossip network (at all)?

## 10.243 Tootsville.Gossip.createConnection

### 10.243.1 Function

Tootsville.Gossip.createConnection is nullary function.

Create and advertise an offer for connection.

WRITEME This deserves better documentation

## 10.244 Tootsville.Gossip.createPacket

### 10.244.1 Function

Tootsville.Gossip.createPacket is a function with lambda list: (c, d, r)

Create and sign a packet.

- c Command to broadcast. This can be the string `logOK`, which is handled as a `_cmd` (in fact, it's the only remaining `_cmd` command); a string beginning with `:`, in which case the command will be a reply packet with `from`; or any other string, in which case it's treated as a `c` command.
- d Data for the command.
- r Optional recipient UUID (or default `$World`)

Automatically adds `a` for author, `u` for user, and `s` signature.

Returns a JSON string of the signed packet.

## 10.245 Tootsville.Gossip.ensureConnected

### 10.245.1 Function

Tootsville.Gossip.ensureConnected is a function with lambda list: (success)

Ensure that we have at least 5 gossip network connections.



## **10.246 Tootsville.Gossip.ensureKeyPair**

### **10.246.1 Function**

Tootsville.Gossip.ensureKeyPair is nullary function.

Ensure that we have an unique public/private key pair for this session

## 10.247 Tootsville.Gossip.gatekeeperAccept

### 10.247.1 Function

Tootsville.Gossip.gatekeeperAccept is a function with lambda list: (peer, event)

Accept an inbound datagram from a peer for an event.

See the server documentation of Section 8.338 [TOOTSVILLE DEFINFINITY], page 592, for a description of the Infinity Mode protocols.

Commands are handled via the Tootsville.Game.Gatekeeper handlers.

## **10.248 Tootsville.Gossip.getICE**

### **10.248.1 Function**

Tootsville.Gossip.getICE is nullary function.

Obtain ICE server info from the game server.

## 10.249 Tootsville.Gossip.getOffer

### 10.249.1 Function

Tootsville.Gossip.getOffer is a function with lambda list: (success)

Accept an offer from the central switchboard

## **10.250 Tootsville.Gossip.openInfinityMode**

### **10.250.1 Function**

Tootsville.Gossip.openInfinityMode is a function with lambda list: (peer, event)

Initiate Infinity mode communications; send a login packet out to \$Eden

## 10.251 Tootsville.Gossip.send

### 10.251.1 Function

Tootsville.Gossip.send is a function with lambda list: (c, d, r, a, v)

Broadcast a packet.

The constructed packet will be signed and broadcast to each of the mesh partners in the gossipnet.

- c is the command;
- d is the command's data (if any);
- r is the target Recipient (originally Room), which defaults to '\$World',
- a is the author (default self), and
- v (via), if present, prevents rebroadcasting the packet to the original sender. V (via) is expected to be null, or an array of UUIDs.

## **10.252 Tootsville.Gossip.sendLogOK**

### **10.252.1 Function**

Tootsville.Gossip.sendLogOK is nullary function.

Send a logOK message to the gossip net.

## 10.253 Tootsville.Gossip.signPacket

### 10.253.1 Function

Tootsville.Gossip.signPacket is a function with lambda list: (c, d, r)

Sign a packet with our private key



## 10.254 Tootsville.Gossip.waitForAnswer

### 10.254.1 Function

Tootsville.Gossip.waitForAnswer is a function with lambda list: (peer, offer, retries, next)

Wait for an answer to an offer which was posted. Comet-type long poll.

## 10.255 Tootsville.GroundBuilder.build

### 10.255.1 Function

Tootsville.GroundBuilder.build is a function with lambda list: (lat, long, alt)

Build the ground plane (terrain map) for the scene at lat, long, alt.

Affects Tootsville.Tank.scene.

## **10.256 Tootsville.GroundBuilder.colorForPlace**

### **10.256.1 Function**

Tootsville.GroundBuilder.colorForPlace is a function with lambda list: (kind)

## **10.257 Tootsville.GroundBuilder.initGroundPlane**

### **10.257.1 Function**

Tootsville.GroundBuilder.initGroundPlane is nullary function.

Initialize the ground plane.

TODO: have a height map across the groundplane.

## **10.258 Tootsville.GroundBuilder.kinds**

### **10.258.1 Variable**

## **10.259 Tootsville.GroundBuilder.paintPlaces**

### **10.259.1 Function**

Tootsville.GroundBuilder.paintPlaces is a function with lambda list: (lat, long, alt)

## **10.260 Tootsville.Login.acceptSignedIn**

### **10.260.1 Function**

Tootsville.Login.acceptSignedIn is a function with lambda list: (result)

    Callback for Firebase completing authentication

## **10.261 Tootsville.Login.addChildFlag**

### **10.261.1 Function**

Tootsville.Login.addChildFlag is a function with lambda list: (li)

Add to LI the child settings flag.

Takes into account if we're in child settings mode or just displaying it.



## 10.262 Tootsville.Login.addChildRequest

### 10.262.1 Function

Tootsville.Login.addChildRequest is a function with lambda list: (li, request)

Add information to a Toot List item about a Child Request.

When the child Toot has an outstanding request, this shows whether it has been granted or denied, and if granted, for how long, and how much of that time remains.

Includes the ability to answer (or change the answer of) the request by triggering a server prompt.

## **10.263 Tootsville.Login.changeSensitivePlayer**

### **10.263.1 Function**

Tootsville.Login.changeSensitivePlayer is a function with lambda list: (button)

Toggle whether the player is marked as Sensitive or not.

## **10.264 Tootsville.Login.childRequestTimeLeft**

### **10.264.1 Function**

Tootsville.Login.childRequestTimeLeft is a function with lambda list: (request)

    Pretty-print the time remaining for a child request

## **10.265 Tootsville.Login.childSettings**

### **10.265.1 Function**

Tootsville.Login.childSettings is nullary function.

Enter the child settings mode.

## **10.266 Tootsville.Login.clearTootsList**

### **10.266.1 Function**

Tootsville.Login.clearTootsList is nullary function.

Clear the login Toots list

## **10.267 Tootsville.Login.considerChildApproval**

### **10.267.1 Function**

Tootsville.Login.considerChildApproval is a function with lambda list: (uuid)

Ask the server to re-prompt us for the Child Request with UUID.

The server will send a "prompt" packet down immediately.

## **10.268 Tootsville.Login.createTootListItem**

### **10.268.1 Function**

Tootsville.Login.createTootListItem is a function with lambda list: (toot)

The set of Toot characters available to the player.

Create a Toot List item for the given Toot JSON object.

## **10.269 Tootsville.Login.dimUnpickedCharacters**

### **10.269.1 Function**

Tootsville.Login.dimUnpickedCharacters is a function with lambda list: (picked)

Dim all the Toot characters other than the one who was PICKED.



## **10.270 Tootsville.Login.disableChildMode**

### **10.270.1 Function**

Tootsville.Login.disableChildMode is a function with lambda list: (name)

Set NAME to no longer be a Child Toot.

## 10.271 Tootsville.Login.doRealLogin

### 10.271.1 Function

Tootsville.Login.doRealLogin is a function with lambda list: (name)

## **10.272 Tootsville.Login.doneEditingSettings**

### **10.272.1 Function**

Tootsville.Login.doneEditingSettings is nullary function.

Leave the Child Settings mode; return to login selection

## **10.273 Tootsville.Login.enableChildMode**

### **10.273.1 Function**

Tootsville.Login.enableChildMode is a function with lambda list: (name)

Set NAME to be a Child Toot.

## **10.274 Tootsville.Login.endLoginMusic**

### **10.274.1 Function**

Tootsville.Login.endLoginMusic is nullary function.

Stop playing the login music and start playing game background music.

## **10.275 Tootsville.Login.fillGoogleUserInfo**

### **10.275.1 Function**

Tootsville.Login.fillGoogleUserInfo is nullary function.

Accept information from Google to fill in the display

## **10.276 Tootsville.Login.findLIForToot**

### **10.276.1 Function**

Tootsville.Login.findLIForToot is a function with lambda list: (name)

Finds the list item representing a Toot named NAME in the login selection list.

## **10.277 Tootsville.Login.finishSignIn**

### **10.277.1 Function**

Tootsville.Login.finishSignIn is a function with lambda list: (idToken)

After signing in, begin choosing a Toot and connect networking.



## **10.278 Tootsville.Login.firebaseLogin**

### **10.278.1 Function**

Tootsville.Login.firebaseLogin is a function with lambda list: (loginPanel)

Start the Firebase login system

## 10.279 Tootsville.Login.generateNewToot

### 10.279.1 Function

Tootsville.Login.generateNewToot is nullary function.

Launch the New Toot panel.

To get the Gossip Parrot prompt first, call Section 10.295 [Tootsville.Login.startCharacterCreation],  
page 2076, instead.

## **10.280 Tootsville.Login.loadTootsList**

### **10.280.1 Function**

Tootsville.Login.loadTootsList is nullary function.

Query the server for my characters after user has signed in.

See Section 8.740 [TOOTSVILLE INFINITY-TOOT-LIST], page 1024,

## **10.281 Tootsville.Login.loginDone**

### **10.281.1 Function**

Tootsville.Login.loginDone is nullary function.

Login has completed; clean up and set up for the game.

## **10.282 Tootsville.Login.loginKidDirty**

### **10.282.1 Function**

Tootsville.Login.loginKidDirty is a function with lambda list: (item)

## **10.283 Tootsville.Login.loginKidDone**

### **10.283.1 Function**

Tootsville.Login.loginKidDone is a function with lambda list: (button)

## **10.284 Tootsville.Login.overlay**

### **10.284.1 Function**

Tootsville.Login.overlay is nullary function.

Open the Login HUD panel

## 10.285 Tootsville.Login.pickCharacter

### 10.285.1 Function

Tootsville.Login.pickCharacter is a function with lambda list: (picked)

Pick “picked” as your Toot to play with today



## **10.286 Tootsville.Login.playWithCharacter**

### **10.286.1 Function**

Tootsville.Login.playWithCharacter is a function with lambda list: (name)

## **10.287 Tootsville.Login.populateTootsList**

### **10.287.1 Function**

Tootsville.Login.populateTootsList is nullary function.

Build the Toots List display from the Toots List in memory.

## **10.288 Tootsville.Login.quit**

### **10.288.1 Function**

Tootsville.Login.quit is nullary function.

## **10.289 Tootsville.Login.removeChildFlag**

### **10.289.1 Function**

Tootsville.Login.removeChildFlag is a function with lambda list: (li)

Remove the Child flag from a LI

## **10.290 Tootsville.Login.saveTootsList**

### **10.290.1 Function**

Tootsville.Login.saveTootsList is a function with lambda list: (list)

Save LIST as the Toots List, then rebuild the display.

## 10.291 Tootsville.Login.serverLinkTokenToCharacter

### 10.291.1 Function

Tootsville.Login.serverLinkTokenToCharacter is a function with lambda list: (character)

Inform the server that we want to play with “character”

See Section 8.718 [TOOTSVILLE INFINITY-PLAY-WITH], page 989,

## **10.292 Tootsville.Login.setSensitiveP**

### **10.292.1 Function**

Tootsville.Login.setSensitiveP is nullary function.

## **10.293 Tootsville.Login.settingsP**

### **10.293.1 Variable**

Is the login panel currently presenting account settings mode?



## **10.294 Tootsville.Login.start**

### **10.294.1 Function**

Tootsville.Login.start is nullary function.

Start the login process

## **10.295 Tootsville.Login.startCharacterCreation**

### **10.295.1 Function**

Tootsville.Login.startCharacterCreation is nullary function.

Start the New Toot creation process.

## **10.296 Tootsville.Login.startSignIn**

### **10.296.1 Function**

Tootsville.Login.startSignIn is nullary function.

Start the login/sign-in process

## **10.297 Tootsville.Login.storeCredentialInfo**

### **10.297.1 Function**

Tootsville.Login.storeCredentialInfo is a function with lambda list: (result)

Get credential information back from Firebase

## **10.298 Tootsville.Login.switchTootsView**

### **10.298.1 Function**

Tootsville.Login.switchTootsView is nullary function.

Show the view for switching Toot characters

## **10.299 Tootsville.Login.toots**

### **10.299.1 Variable**

The Toots List in memory.

## **10.300 Tootsville.Login.updateNote**

### **10.300.1 Function**

Tootsville.Login.updateNote is a function with lambda list: (tootName, event)

Update the note attached to tootName

An event handler to be connected to the TEXTAREA control.

## **10.301 Tootsville.Login.validChildCode**

### **10.301.1 Function**

Tootsville.Login.validChildCode is a function with lambda list: (string)

Determines whether STRING might be a valid Child Code.



## **10.302 Tootsville.SceneBuilder.addFurn**

### **10.302.1 Function**

Tootsville.SceneBuilder.addFurn is a function with lambda list: (item)

## **10.303 Tootsville.SceneBuilder.addItem1**

### **10.303.1 Function**

Tootsville.SceneBuilder.addItem1 is a function with lambda list: (item)

## 10.304 Tootsville.SceneBuilder.addItem2

### 10.304.1 Function

Tootsville.SceneBuilder.addItem2 is a function with lambda list: (item)

Adds a furniture item in the “itm2” format

See Section 8.770 [TOOTSVILLE ITEM-INFO], page 1055,

```
{ uuid:  
position: { x: y: z: },  
facing: radians,  
baseColor: color,  
altColor: color,  
energy: number,  
scale: { x: y: z: },  
world: { world: lat: long: alt: },  
template:  
{ id:  
name:  
description:  
trade: [ Y N X ],  
avatar:  
energyKind:  
energyMax:  
onZero:  
wearSlot:  
weight: } }
```

## **10.305 Tootsville.SceneBuilder.addPlace**

### **10.305.1 Function**

Tootsville.SceneBuilder.addPlace is a function with lambda list: (key, info)

## **10.306 Tootsville.SceneBuilder.addText**

### **10.306.1 Function**

Tootsville.SceneBuilder.addText is a function with lambda list: (item)

## 10.307 Tootsville.SceneBuilder.build

### 10.307.1 Function

Tootsville.SceneBuilder.build is a function with lambda list: (x, y, z)

Place furniture as found in the current scene.

Affects Tootsville.Tank.scene

## 10.308 Tootsville.SkyBuilder.build

### 10.308.1 Function

Tootsville.SkyBuilder.build is a function with lambda list: (world)

Build the sky for the current environment. Reads the sky values at Tootsville.SkyBuilder.sky and affects the scene at Tootsville.Tank.scene.

Depending on the world in question, the sky may have these layers:

1. The first layer is the base color of the sky. On Choerogryllum, this varies from a blue bordering on white at noon (09:00) to pitch black at midnight (00:00) and should have redness applied at sunset. The ambient color lighting of the scene is likewise affected. In space, it is always black.
2. The second layer is a starfield, which fades with the sun's altitude.
3. The third layer has the sun.
4. The fourth layer has The Moon, The Other Moon, and the Pink Moon, as appropriate, and Choerogryllum, when in space.
5. The fifth layer is cloud cover (when on the planet).
6. Finally, the sixth layer is any precipitation effect.

## **10.309 Tootsville.SkyBuilder.buildMatchingSky**

### **10.309.1 Function**

Tootsville.SkyBuilder.buildMatchingSky is a function with lambda list: (sky)



## **10.310 Tootsville.SkyBuilder.buildMatchingWeather**

### **10.310.1 Function**

Tootsville.SkyBuilder.buildMatchingWeather is a function with lambda list: (weather)

## **10.311 Tootsville.SkyBuilder.setCloudCover**

### **10.311.1 Function**

Tootsville.SkyBuilder.setCloudCover is nullary function.

Set up clouds above the terrain based on the map from the server. Only used on Chærogyryllum.

## 10.312 Tootsville.SkyBuilder.setFirstSkyLayer

### 10.312.1 Function

Tootsville.SkyBuilder.setFirstSkyLayer is a function with lambda list: (atmosphereP)

The first layer of the sky is the ambient light of the scene, and the color of the sky itself. This is based entirely upon the sun position.

This is always black in space (when atmosphereP is false).

The sky data is taken from Tootsville.SkyBuilder.sky and the scene is Tootsville.Tank.scene.

XXX Some day, using a GLSL shader for the background would be awesome, but that's more work than is valuable at this stage (for BRP). If some volunteer is excited about the notion, it would be a very nice touch. Keep in mind that the 18 hour day and 360 day year will require certain alterations to any stock routines based upon a 24 hour day and 365.2489 day year.

## **10.313 Tootsville.SkyBuilder.setMoon**

### **10.313.1 Function**

Tootsville.SkyBuilder.setMoon is a function with lambda list: (whichMoon)

Position one of the moons relative to the viewer. The moon in question's identity is passed in.

## **10.314 Tootsville.SkyBuilder.setPlanet**

### **10.314.1 Function**

Tootsville.SkyBuilder.setPlanet is nullary function.

Position the planet Chærogryllum relative to the viewer. Used when the player is in orbit or on one of the moons.

## **10.315 Tootsville.SkyBuilder.setPrecipitation**

### **10.315.1 Function**

Tootsville.SkyBuilder.setPrecipitation is nullary function.

Set precipitation, if any. Only used on Chærogryllum.

## 10.316 Tootsville.SkyBuilder.setStarfield

### 10.316.1 Function

Tootsville.SkyBuilder.setStarfield is a function with lambda list: (atmosphereP)

The second layer of the sky are the stars, which are faded with the relative brightness of the background layer. ie: We actually reduce the opacity of the stars when the sun is up.  
FIXME

## **10.317 Tootsville.SkyBuilder.setSun**

### **10.317.1 Function**

Tootsville.SkyBuilder.setSun is nullary function.

Position the sun relative to the viewer



## 10.318 Tootsville.SkyBuilder.setTheMoon

### 10.318.1 Function

Tootsville.SkyBuilder.setTheMoon is nullary function.

Position The Moon relative to the viewer. See Section 10.313 [Tootsville.SkyBuilder.setMoon], page 2094.

## **10.319 Tootsville.SkyBuilder.setTheOtherMoon**

### **10.319.1 Function**

Tootsville.SkyBuilder.setTheOtherMoon is nullary function.

Position The Other Moon relative to the viewer. See Section 10.313 [Tootsville.SkyBuilder.setMoon], page 2094.

## 10.320 Tootsville.SkyBuilder.setThePinkMoon

### 10.320.1 Function

Tootsville.SkyBuilder.setThePinkMoon is nullary function.

Position The Pink Moon relative to the viewer. See Section 10.313 [Tootsville.SkyBuilder.setMoon], page 2094.

## **10.321 Tootsville.SkyBuilder.sunX**

### **10.321.1 Function**

Tootsville.SkyBuilder.sunX is nullary function.

Get the instantaneous position of the sun in X

Accurate to  $\pm 1$  min (of time)

## **10.322 Tootsville.SkyBuilder.sunY**

### **10.322.1 Function**

Tootsville.SkyBuilder.sunY is nullary function.

Get the instantaneous position of the sun in Y

Accurate to  $\pm 1$  min (of time)

## **10.323 Tootsville.SkyBuilder.update**

### **10.323.1 Function**

Tootsville.SkyBuilder.update is a function with lambda list: (world)

Update sky positions and the like.

XXX Some things aren't able to be updated yet.

## **10.324 Tootsville.SkyBuilder.updateSkyData**

### **10.324.1 Function**

Tootsville.SkyBuilder.updateSkyData is nullary function.

Fetch sky data from the game server

Updates Tootsville.SkyBuilder.sky

## 10.325 Tootsville.Tank.CameraManager.positionCameraForAvatarCloseUp

### 10.325.1 Function

Tootsville.Tank.CameraManager.positionCameraForAvatarCloseUp is a function with lambda list: (camera, avatar)

Position the camera to have the avatar about  $\frac{1}{4}$  the screen width.



## 10.326 Tootsville.Tank.CameraManager.positionCameraForAvatarViewer

### 10.326.1 Function

Tootsville.Tank.CameraManager.positionCameraForAvatarViewer is a function with lambda list: (camera, avatar)

Position the camera to enclose the avatar completely.

## 10.327 Tootsville.Tank.CameraManager.positionCameraForGameBoard

### 10.327.1 Function

Tootsville.Tank.CameraManager.positionCameraForGameBoard is a function with lambda list: (camera, avatar)

Position the camera to have the avatar about 120 the screen width.

## **10.328 Tootsville.Tank.afterRender**

### **10.328.1 Function**

Tootsville.Tank.afterRender is nullary function.

This event handler is called whenever a frame in the 3D scene has been rendered.

## **10.329 Tootsville.Tank.attachmentOverlaysNeedUpdateP**

### **10.329.1 Variable**

Indicates whether the 2D overlay attachments need updating.

When true, the scene has changed in some way that may invalidate the positions of things like speech balloons.

XXX This is currently ignored, and we always update the attachments on every frame.

## **10.330 Tootsville.Tank.createScene**

### **10.330.1 Function**

Tootsville.Tank.createScene is nullary function.

Create the text scene with ground plane and the player's Toot with a static light.

## **10.331 Tootsville.Tank.destroyAvatar**

### **10.331.1 Function**

Tootsville.Tank.destroyAvatar is a function with lambda list: (avatar)

Destroy avatar and attachments.

## 10.332 Tootsville.Tank.findAvatar

### 10.332.1 Function

Tootsville.Tank.findAvatar is a function with lambda list: (avatarName)

Find an avatar by name; may return null if we don't know about that avatar yet.

## **10.333 Tootsville.Tank.getCanvas**

### **10.333.1 Function**

Tootsville.Tank.getCanvas is nullary function.

Find or create the CANVAS object onto which the 3D scene is rendered.



## 10.334 Tootsville.Tank.getLargestChildMesh

### 10.334.1 Function

Tootsville.Tank.getLargestChildMesh is a function with lambda list: (object)

Find the child mesh with the greatest volume.

If there are no children, returns the parent mesh. Otherwise, always returns a child.  
Uses the radius of the bounding sphere as a proxy for volume computations.

## **10.335 Tootsville.Tank.init3DEngine**

### **10.335.1 Function**

Tootsville.Tank.init3DEngine is nullary function.

Initialize the 3D engine, including Babylon 3D.

The main entry point is Section 10.344 [Tootsville.Tank.start3D], page 2125, which eventually invokes this. This function actually connects the 3D engine to the CANVAS object and 2D event system.

## **10.336 Tootsville.Tank.initArcCamera**

### **10.336.1 Function**

Tootsville.Tank.initArcCamera is nullary function.

## **10.337 Tootsville.Tank.initOTSCamera**

### **10.337.1 Function**

Tootsville.Tank.initOTSCamera is nullary function.

Initialize the Over-The-Shoulder camera.

This is the main follow camera for the game. This camera follows the player's Toot through the scene.

## **10.338 Tootsville.Tank.initPlayerToot**

### **10.338.1 Function**

Tootsville.Tank.initPlayerToot is nullary function.

Initialize our local player's Toot object.

We know that it, at least, will always exist.

## **10.339 Tootsville.Tank.initScene**

### **10.339.1 Function**

Tootsville.Tank.initScene is nullary function.

Initialize the Babylon 3D scene object.

## **10.340 Tootsville.Tank.loadUISounds**

### **10.340.1 Function**

Tootsville.Tank.loadUISounds is nullary function.

    Enqueue some foley sound effects that will be used in the scene.

## **10.341 Tootsville.Tank.playerAvatar**

### **10.341.1 Function**

Tootsville.Tank.playerAvatar is nullary function.

The avatar for the active local player



## **10.342 Tootsville.Tank.prepareFor3D**

### **10.342.1 Function**

Tootsville.Tank.prepareFor3D is nullary function.

Prepare the libraries needed for the 3D scene (Babylon.js).

We can load these hefty libraries asynchronously whilst the player is busy signing in.

## **10.343 Tootsville.Tank.shutDown**

### **10.343.1 Function**

Tootsville.Tank.shutDown is nullary function.

Shut down the 3D environment cleanly.

## **10.344 Tootsville.Tank.start3D**

### **10.344.1 Function**

Tootsville.Tank.start3D is nullary function.

Start the 3D engine, after doing any necessary preparatory work. This is the main entry point for the 3D simulation engine.

## **10.345 Tootsville.Tank.start3DReal**

### **10.345.1 Function**

Tootsville.Tank.start3DReal is nullary function.

Ensure that all libraries are loaded and actually start the 3D engine. Called by Section 10.344 [Tootsville.Tank.start3D], page 2125,

## **10.346 Tootsville.Tank.startRenderLoop**

### **10.346.1 Function**

Tootsville.Tank.startRenderLoop is nullary function.

Start the 3D render loop running.

## 10.347 Tootsville.Tank.updateAvatarFor

### 10.347.1 Function

Tootsville.Tank.updateAvatarFor is a function with lambda list: (avatarName)

Update the avatar model for avatarName, by looking up its description Tootsville.Tank.avatars.

## **10.348 Tootsville.Tank.updateCamera**

### **10.348.1 Function**

Tootsville.Tank.updateCamera is nullary function.

Reposition the camera as needed to track the player

**10.349 Tootsville.UI.Audio.context****10.349.1 Variable**



## **10.350 Tootsville.UI.Audio.gainNode**

### **10.350.1 Variable**

## **10.351 Tootsville.UI.Audio.setVolume**

### **10.351.1 Function**

Tootsville.UI.Audio.setVolume is a function with lambda list: (newVolume)

Set the volume to “newVolume”%

## **10.352 Tootsville.UI.Audio.updateVolumeMuteIcon**

### **10.352.1 Function**

Tootsville.UI.Audio.updateVolumeMuteIcon is nullary function.

Update the volume Mute indicator in the control panel

## **10.353 Tootsville.UI.Audio.updateVolumeSlider**

### **10.353.1 Function**

Tootsville.UI.Audio.updateVolumeSlider is nullary function.

Update the volume slider in the control panel.

## **10.354 Tootsville.UI.Audio.updateVolumeUI**

### **10.354.1 Function**

Tootsville.UI.Audio.updateVolumeUI is nullary function.

Update the volume controls UI in the control panel menu

## **10.355 Tootsville.UI.Audio.volumeDown**

### **10.355.1 Function**

Tootsville.UI.Audio.volumeDown is nullary function.

Lower the volume by 10%, down to a mininum of 0.

## **10.356 Tootsville.UI.Audio.volumeMute**

### **10.356.1 Function**

Tootsville.UI.Audio.volumeMute is nullary function.

Temporarily mute or unmute the volume.

## **10.357 Tootsville.UI.Audio.volumeUp**

### **10.357.1 Function**

Tootsville.UI.Audio.volumeUp is nullary function.

Raise the volume by 10%, up to a maximum of 100%.



## **10.358 Tootsville.UI.Gamepad.ROTATION\_SPEED**

### **10.358.1 Variable**

## 10.359 Tootsville.UI.Gamepad.addGamepad

### 10.359.1 Function

Tootsville.UI.Gamepad.addGamepad is a function with lambda list: (gamepad)

Add a gamepad and initialize state data.

## **10.360 Tootsville.UI.Gamepad.axisUpdate**

### **10.360.1 Function**

Tootsville.UI.Gamepad.axisUpdate is a function with lambda list: (controllerIndex)

## **10.361 Tootsville.UI.Gamepad.buttonEvent**

### **10.361.1 Function**

Tootsville.UI.Gamepad.buttonEvent is a function with lambda list: (controllerIndex, buttonIndex, value)

## **10.362 Tootsville.UI.Gamepad.connectHandler**

### **10.362.1 Function**

Tootsville.UI.Gamepad.connectHandler is a function with lambda list: (ev)

Event handler for gamepad connections

## **10.363 Tootsville.UI.Gamepad.controllerState**

### **10.363.1 Variable**

Gamepad controller state data.

TODO, document format

## **10.364 Tootsville.UI.Gamepad.controllers**

### **10.364.1 Variable**

All connected gamepad controllers.

## **10.365 Tootsville.UI.Gamepad.disconnectHandler**

### **10.365.1 Function**

Tootsville.UI.Gamepad.disconnectHandler is a function with lambda list: (e)

Event handler for gamepad disconnections.



## **10.366 Tootsville.UI.Gamepad.removeGamepad**

### **10.366.1 Function**

Tootsville.UI.Gamepad.removeGamepad is a function with lambda list: (gamepad)

Remove a gamepad from the active state.

## **10.367 Tootsville.UI.Gamepad.scanGamepads**

### **10.367.1 Function**

Tootsville.UI.Gamepad.scanGamepads is nullary function.

Scan gamepads for updates

## **10.368 Tootsville.UI.Gamepad.updateStatus**

### **10.368.1 Function**

Tootsville.UI.Gamepad.updateStatus is nullary function.

Update gamepad status.

## **10.369 Tootsville.UI.HUD.beginWatchingPaperdollWindowForClose**

### **10.369.1 Function**

Tootsville.UI.HUD.beginWatchingPaperdollWindowForClose is nullary function.

Watch the paperdoll (large) window for Close events (ie, layer is made invisible or hidden in the DOM).

## 10.370 Tootsville.UI.HUD.clickedOnMesh

### 10.370.1 Function

Tootsville.UI.HUD.clickedOnMesh is a function with lambda list: (mesh, picked)

Respond to a user click (tap) on a mesh in the tank

## **10.371 Tootsville.UI.HUD.closePanel**

### **10.371.1 Function**

Tootsville.UI.HUD.closePanel is nullary function.

Close (hide) the active HUD panel.

## **10.372 Tootsville.UI.HUD.closeTalkBox**

### **10.372.1 Function**

Tootsville.UI.HUD.closeTalkBox is a function with lambda list: (event)

Close (hide) the Talk Box

## **10.373 Tootsville.UI.HUD.connectTalkBox**

### **10.373.1 Function**

Tootsville.UI.HUD.connectTalkBox is nullary function.

Connect events for the Talk box widgets at the bottom of the display.



## 10.374 Tootsville.UI.HUD.convertCanvasEventTo3D

### 10.374.1 Function

Tootsville.UI.HUD.convertCanvasEventTo3D is a function with lambda list: (event)

Convert an event on the HUD or CANVAS object into a 3D event as appropriate.

## **10.375 Tootsville.UI.HUD.createHUDLoaderPanel**

### **10.375.1 Function**

Tootsville.UI.HUD.createHUDLoaderPanel is a function with lambda list: (panel)

Create a placeholder “loading” pop-up for a HUD panel.

## **10.376 Tootsville.UI.HUD.createPaperdollCanvas**

### **10.376.1 Function**

Tootsville.UI.HUD.createPaperdollCanvas is a function with lambda list: (paperdoll)

Create the canvas for paperdoll display

## **10.377 Tootsville.UI.HUD.destroyHUD**

### **10.377.1 Function**

Tootsville.UI.HUD.destroyHUD is nullary function.

Destroy the HUD layer.

## **10.378 Tootsville.UI.HUD.dropHUDPanels**

### **10.378.1 Function**

Tootsville.UI.HUD.dropHUDPanels is nullary function.

Drop all HUD panels to force reloading them. May not always work due to caching.

## **10.379 Tootsville.UI.HUD.getOpenPanel**

### **10.379.1 Function**

Tootsville.UI.HUD.getOpenPanel is nullary function.

Get the name of the currently-visible HUD panel.

## **10.380 Tootsville.UI.HUD.initHUD**

### **10.380.1 Function**

Tootsville.UI.HUD.initHUD is nullary function.

Set up the HUD layer and start housekeeping.

## **10.381 Tootsville.UI.HUD.loadHTML**

### **10.381.1 Function**

Tootsville.UI.HUD.loadHTML is a function with lambda list: (src)

Load an HTML layer into a container.



## 10.382 Tootsville.UI.HUD.loadHUDPanel

### 10.382.1 Function

Tootsville.UI.HUD.loadHUDPanel is a function with lambda list: (panelName, finish)

Load a HUD Panel from /play/UI/panels/. Each panel has an HTML and a Javascript component associated with it.

## **10.383 Tootsville.UI.HUD.loadScriptIntoDiv**

### **10.383.1 Function**

Tootsville.UI.HUD.loadScriptIntoDiv is a function with lambda list: (src, div)

Load a SCRIPT into a given DIV container.

## **10.384 Tootsville.UI.HUD.nameTagClicked**

### **10.384.1 Function**

Tootsville.UI.HUD.nameTagClicked is a function with lambda list: (event)

Respond to a user click (tap) on a name tag

## **10.385 Tootsville.UI.HUD.openPaperdoll**

### **10.385.1 Function**

Tootsville.UI.HUD.openPaperdoll is a function with lambda list: (event)

Open the Paperdoll display from the paperdoll-mini widget.

## **10.386 Tootsville.UI.HUD.openTalkBox**

### **10.386.1 Function**

Tootsville.UI.HUD.openTalkBox is a function with lambda list: (event)

Open (reveal) the Talk Box.

## **10.387 Tootsville.UI.HUD.paperdollCurrentP**

### **10.387.1 Function**

Tootsville.UI.HUD.paperdollCurrentP is nullary function.

Is the paperdoll mini up-to-date?

## 10.388 Tootsville.UI.HUD.positionPaperdollMini

### 10.388.1 Function

Tootsville.UI.HUD.positionPaperdollMini is nullary function.

Position the paperdoll appropriately for the stage box in the large paperdoll display, or the mini box (widget) in the lower-right corner.

## **10.389 Tootsville.UI.HUD.refreshAttachmentOverlays**

### **10.389.1 Function**

Tootsville.UI.HUD.refreshAttachmentOverlays is nullary function.

Refresh all 2D attachment overlays to follow the 3D scene.



## **10.390 Tootsville.UI.HUD.refreshAttachmentsForAvatar**

### **10.390.1 Function**

Tootsville.UI.HUD.refreshAttachmentsForAvatar is a function with lambda list: (avatar)

Refresh the 2D attachments for one avatar.

## **10.391 Tootsville.UI.HUD.refreshEquipment**

### **10.391.1 Function**

Tootsville.UI.HUD.refreshEquipment is nullary function.

Refresh the display of the active equipment item.

## 10.392 Tootsville.UI.HUD.refreshHUD

### 10.392.1 Function

Tootsville.UI.HUD.refreshHUD is nullary function.

Refresh HUD elements that are set by server events (other than the clock), such as equipment, talk status, and wallet readouts.

These elements are refreshed about every 333 ms, but occur in a 4msec (the minimum allowed setTimeout value for HTML5) timeout handlers so as to be more effectively asynchronous (and because browsers bitch if you run too long in one setInterval handler).

## 10.393 Tootsville.UI.HUD.refreshNameTagAttachment

### 10.393.1 Function

Tootsville.UI.HUD.refreshNameTagAttachment is a function with lambda list: (model, nameTag)

Refresh one 2D name tag attachment object.

These attachments need to be refreshed to keep in sync with the underlying 3D scene from time to time.

## **10.394 Tootsville.UI.HUD.refreshPaperdoll**

### **10.394.1 Function**

Tootsville.UI.HUD.refreshPaperdoll is nullary function.

Ensure that the paperdoll is up-to-date, updating it if needed.

## **10.395 Tootsville.UI.HUD.refreshSpeechAttachment**

### **10.395.1 Function**

Tootsville.UI.HUD.refreshSpeechAttachment is a function with lambda list: (model, speechBubble)

Refresh one 2D speech attachment object.

These attachments need to be refreshed to keep in sync with the underlying 3D scene from time to time.

## **10.396 Tootsville.UI.HUD.refreshTalkStatus**

### **10.396.1 Function**

Tootsville.UI.HUD.refreshTalkStatus is nullary function.

Refresh the status of the Talk Box (disconnected, sensitive, or regular).

## **10.397 Tootsville.UI.HUD.refreshTimeLeft**

### **10.397.1 Function**

Tootsville.UI.HUD.refreshTimeLeft is nullary function.

Refresh the time remaining indicator for a child player



## **10.398 Tootsville.UI.HUD.refreshWallet**

### **10.398.1 Function**

Tootsville.UI.HUD.refreshWallet is nullary function.

Refresh the wallet display, both in the HUD and (if loaded) the Wallet app in Tootnix.

## **10.399 Tootsville.UI.HUD.returnPaperdollMini**

### **10.399.1 Function**

Tootsville.UI.HUD.returnPaperdollMini is nullary function.

Return the paperdoll from the large window to the icon widget.

## 10.400 Tootsville.UI.HUD.setPaperdollForPlayerAvatar

### 10.400.1 Function

Tootsville.UI.HUD.setPaperdollForPlayerAvatar is a function with lambda list: (paperdoll)

Set the paperdoll avatar values to the current avatar values, and also remember its current height.

If any of these changes, the paperdoll will need to be redrawn. See Section 10.387 [Tootsville.UI.HUD.paperdollCurrentP], page 2168.

## **10.401 Tootsville.UI.HUD.showCamera**

### **10.401.1 Function**

Tootsville.UI.HUD.showCamera is nullary function.

Show the camera widget

## **10.402 Tootsville.UI.HUD.showControlPanel**

### **10.402.1 Function**

Tootsville.UI.HUD.showControlPanel is nullary function.

Show the control panel menu

## **10.403 Tootsville.UI.HUD.showHUDPanel**

### **10.403.1 Function**

Tootsville.UI.HUD.showHUDPanel is a function with lambda list: (panel, div)

Show the HUD panel named in the given DIV container.

## **10.404 Tootsville.UI.HUD.showMobile**

### **10.404.1 Function**

Tootsville.UI.HUD.showMobile is nullary function.

Show the player's mobile device

## **10.405 Tootsville.UI.HUD.showPlayerCard**

### **10.405.1 Function**

Tootsville.UI.HUD.showPlayerCard is a function with lambda list: (name)

Show the Player Card pop-up for another player



## **10.406 Tootsville.UI.HUD.switchActiveItem**

### **10.406.1 Function**

Tootsville.UI.HUD.switchActiveItem is nullary function.

Switch the active item with the secondary item.

**10.407 Tootsville.UI.HUD.talkBoxOpenP****10.407.1 Variable**

If true, the Talk Box is open (visible).

## 10.408 Tootsville.UI.HUD.toggleElement

### 10.408.1 Function

Tootsville.UI.HUD.toggleElement is a function with lambda list: (element)

Toggle whether ELEMENT is displayed or not (with a transition fade).

## **10.409 Tootsville.UI.HUD.toggleHUDPanel**

### **10.409.1 Function**

Tootsville.UI.HUD.toggleHUDPanel is a function with lambda list: (panel)

Toggle the visibility of the named HUD panel.

## **10.410 Tootsville.UI.HUD.toggleTalkBox**

### **10.410.1 Function**

Tootsville.UI.HUD.toggleTalkBox is nullary function.

Toggle visibility of the Talk Box

## **10.411 Tootsville.UI.HUD.toggleTalkEmoji**

### **10.411.1 Function**

Tootsville.UI.HUD.toggleTalkEmoji is a function with lambda list: (event)

Toggle visibility of the Emoji selector for the Talk Box.

## **10.412 Tootsville.UI.HUD.toggleTalkExpression**

### **10.412.1 Function**

Tootsville.UI.HUD.toggleTalkExpression is a function with lambda list: (event)

Toggle visibility of the Expressions selector for the Talk Box.

## **10.413 Tootsville.UI.HUD.toggleTalkLoud**

### **10.413.1 Function**

Tootsville.UI.HUD.toggleTalkLoud is a function with lambda list: (event)

Toggle visibility of the Loudness selector for the Talk Box.



## **10.414 Tootsville.UI.Keys.arrowDown**

### **10.414.1 Function**

Tootsville.UI.Keys.arrowDown is a function with lambda list: (event)

## **10.415 Tootsville.UI.Keys.arrowLeft**

### **10.415.1 Function**

Tootsville.UI.Keys.arrowLeft is a function with lambda list: (event)

## **10.416 Tootsville.UI.Keys.arrowRight**

### **10.416.1 Function**

Tootsville.UI.Keys.arrowRight is a function with lambda list: (event)

## **10.417 Tootsville.UI.Keys.arrowUp**

### **10.417.1 Function**

Tootsville.UI.Keys.arrowUp is a function with lambda list: (event)

## **10.418 Tootsville.UI.Keys.backwardChar**

### **10.418.1 Function**

Tootsville.UI.Keys.backwardChar is a function with lambda list: (event)

Move the cursor backwards one character in the speaking box.

## **10.419 Tootsville.UI.Keys.backwardSentence**

### **10.419.1 Function**

Tootsville.UI.Keys.backwardSentence is a function with lambda list: (event)

Move the cursor back to the previous sentence start.

Sentences are defined to be delimited by period, exclamation point, or question mark.

## **10.420 Tootsville.UI.Keys.backwardWord**

### **10.420.1 Function**

Tootsville.UI.Keys.backwardWord is a function with lambda list: (event)

Move the cursor back one word.

A word is considered to consist of contiguous letters or digits.

## **10.421 Tootsville.UI.Keys.beginShouting**

### **10.421.1 Function**

Tootsville.UI.Keys.beginShouting is a function with lambda list: (event)

Set the speaking volume to shouting.



## 10.422 Tootsville.UI.Keys.beginSpeaking

### 10.422.1 Function

Tootsville.UI.Keys.beginSpeaking is a function with lambda list: (event)

Set the speaking volume to speaking normally (neither shouting nor whispering)

## **10.423 Tootsville.UI.Keys.beginWhispering**

### **10.423.1 Function**

Tootsville.UI.Keys.beginWhispering is a function with lambda list: (event)

Set the speaking volume to whispering.

## **10.424 Tootsville.UI.Keys.beginningOfLine**

### **10.424.1 Function**

Tootsville.UI.Keys.beginningOfLine is a function with lambda list: (event)

Move the cursor to the start of the line.

## **10.425 Tootsville.UI.Keys.capitalizeWord**

### **10.425.1 Function**

Tootsville.UI.Keys.capitalizeWord is a function with lambda list: (event)

Upcase the first letter of the word under the cursor, and downcase the remaining letters.

## **10.426 Tootsville.UI.Keys.deleteBackwardChar**

### **10.426.1 Function**

Tootsville.UI.Keys.deleteBackwardChar is a function with lambda list: (event)

Delete the character to the left of the cursor.

## **10.427 Tootsville.UI.Keys.deleteChar**

### **10.427.1 Function**

Tootsville.UI.Keys.deleteChar is a function with lambda list: (event)

Delete the character to the right of the cursor.

## **10.428 Tootsville.UI.Keys.downcaseWord**

### **10.428.1 Function**

Tootsville.UI.Keys.downcaseWord is a function with lambda list: (event)

Downcase the word under the cursor.

## **10.429 Tootsville.UI.Keys.endOfLine**

### **10.429.1 Function**

Tootsville.UI.Keys.endOfLine is a function with lambda list: (event)

Move the cursor to after the end of the line.



## 10.430 Tootsville.UI.Keys.executeExtendedCommand

### 10.430.1 Function

Tootsville.UI.Keys.executeExtendedCommand is a function with lambda list: (event)

Reserved for future use.

Should prompt for an extended command to execute. This is bound to M-x and is analogous to the function in Emacs.

## **10.431 Tootsville.UI.Keys.forwardChar**

### **10.431.1 Function**

Tootsville.UI.Keys.forwardChar is a function with lambda list: (event)

Move the cursor forward one character.

## **10.432 Tootsville.UI.Keys.forwardSentence**

### **10.432.1 Function**

Tootsville.UI.Keys.forwardSentence is a function with lambda list: (event)

## **10.433 Tootsville.UI.Keys.forwardWord**

### **10.433.1 Function**

Tootsville.UI.Keys.forwardWord is a function with lambda list: (event)

## **10.434 Tootsville.UI.Keys.help**

### **10.434.1 Function**

Tootsville.UI.Keys.help is a function with lambda list: (event)

Load the game help panel

## **10.435 Tootsville.UI.Keys.insertChar**

### **10.435.1 Function**

Tootsville.UI.Keys.insertChar is a function with lambda list: (event)

## **10.436 Tootsville.UI.Keys.isearch**

### **10.436.1 Function**

Tootsville.UI.Keys.isearch is a function with lambda list: (event)

## **10.437 Tootsville.UI.Keys.isearchBackward**

### **10.437.1 Function**

Tootsville.UI.Keys.isearchBackward is a function with lambda list: (event)



## **10.438 Tootsville.UI.Keys.keyboardQuit**

### **10.438.1 Function**

Tootsville.UI.Keys.keyboardQuit is a function with lambda list: (event)

## **10.439 Tootsville.UI.Keys.killLine**

### **10.439.1 Function**

Tootsville.UI.Keys.killLine is a function with lambda list: (event)

Delete the entire contents of the speaking box.

## **10.440 Tootsville.UI.Keys.killRegion**

### **10.440.1 Function**

Tootsville.UI.Keys.killRegion is a function with lambda list: (event)

## **10.441 Tootsville.UI.Keys.killRingSave**

### **10.441.1 Function**

Tootsville.UI.Keys.killRingSave is a function with lambda list: (event)

## **10.442 Tootsville.UI.Keys.killSentence**

### **10.442.1 Function**

Tootsville.UI.Keys.killSentence is a function with lambda list: (event)

## **10.443 Tootsville.UI.Keys.killWord**

### **10.443.1 Function**

Tootsville.UI.Keys.killWord is a function with lambda list: (event)

## **10.444 Tootsville.UI.Keys.nextHistoryLine**

### **10.444.1 Function**

Tootsville.UI.Keys.nextHistoryLine is a function with lambda list: (event)

## **10.445 Tootsville.UI.Keys.prefixCc**

### **10.445.1 Function**

Tootsville.UI.Keys.prefixCc is a function with lambda list: (event)



## **10.446 Tootsville.UI.Keys.prefixCx**

### **10.446.1 Function**

Tootsville.UI.Keys.prefixCx is a function with lambda list: (event)

## **10.447 Tootsville.UI.Keys.priorHistoryLine**

### **10.447.1 Function**

Tootsville.UI.Keys.priorHistoryLine is a function with lambda list: (event)

## **10.448 Tootsville.UI.Keys.selectAll**

### **10.448.1 Function**

Tootsville.UI.Keys.selectAll is a function with lambda list: (event)

## **10.449 Tootsville.UI.Keys.speakLine**

### **10.449.1 Function**

Tootsville.UI.Keys.speakLine is a function with lambda list: (event)

## **10.450 Tootsville.UI.Keys.textEntry**

### **10.450.1 Function**

Tootsville.UI.Keys.textEntry is a function with lambda list: (event)

## **10.451 Tootsville.UI.Keys.transposeChars**

### **10.451.1 Function**

Tootsville.UI.Keys.transposeChars is a function with lambda list: (event)

Switch the characters before and after the cursor

## **10.452 Tootsville.UI.Keys.transposeWords**

### **10.452.1 Function**

Tootsville.UI.Keys.transposeWords is a function with lambda list: (event)

Switch the word under the cursor with the prior word.

When on a non-word character, switch the words before and after it.

## **10.453 Tootsville.UI.Keys.upcaseWord**

### **10.453.1 Function**

Tootsville.UI.Keys.upcaseWord is a function with lambda list: (event)



## **10.454 Tootsville.UI.Keys.yank**

### **10.454.1 Function**

Tootsville.UI.Keys.yank is a function with lambda list: (event)

## **10.455 Tootsville.UI.Keys.yankPop**

### **10.455.1 Function**

Tootsville.UI.Keys.yankPop is a function with lambda list: (event)

## **10.456 Tootsville.UI.WaWa.build**

### **10.456.1 Function**

Tootsville.UI.WaWa.build is a function with lambda list: (phrase, finish)

## **10.457 Tootsville.UI.WaWa.playChained**

### **10.457.1 Function**

Tootsville.UI.WaWa.playChained is a function with lambda list: (chain, finish)

## 10.458 Tootsville.UI.WaWa.playShifted

### 10.458.1 Function

Tootsville.UI.WaWa.playShifted is a function with lambda list: (file, speed=1, after)

Play a sound sample pitch-shifted by the speed difference given.

## **10.459 Tootsville.UI.WaWa.stop**

### **10.459.1 Function**

Tootsville.UI.WaWa.stop is a function with lambda list: (source)

## 10.460 Tootsville.UI.clickedOnItem

### 10.460.1 Function

Tootsville.UI.clickedOnItem is a function with lambda list: (meshName, picked)

WRITEME — this function is not yet documented.

## **10.461 Tootsville.UI.commands**

### **10.461.1 Variable**



## 10.462 Tootsville.UI.confirmPretty

### 10.462.1 Function

Tootsville.UI.confirmPretty is a function with lambda list: (title,text,accept)

Present a nice UI box to confirm whether to do something or not.

The title and text are displayed. The “accept” text is displayed on one button; the negative button will read “Cancel” unless the “accept” text reads “Yes,” in which case it will read “No.”

## **10.463 Tootsville.UI.findAdjacentEntity**

### **10.463.1 Function**

Tootsville.UI.findAdjacentEntity is nullary function.

Discover the nearest entity within “arms’ reach” of the player’s facing direction.

This is for e.g. game pad or keyboard inputs.

## **10.464 Tootsville.UI.forceQuit**

### **10.464.1 Function**

Tootsville.UI.forceQuit is nullary function.

Quit without prompting

## **10.465 Tootsville.UI.insertEmoji**

### **10.465.1 Function**

Tootsville.UI.insertEmoji is a function with lambda list: (event)

## **10.466 Tootsville.UI.interact**

### **10.466.1 Function**

Tootsville.UI.interact is a function with lambda list: (entity)

WRITEME — this function is not yet documented.

## **10.467 Tootsville.UI.makeDivOrParagraph**

### **10.467.1 Function**

Tootsville.UI.makeDivOrParagraph is a function with lambda list: (text)

Turns a string without HTML into a paragraph, one containing HTML markup into a DIV.

## 10.468 Tootsville.UI.makeIDFromTitle

### 10.468.1 Function

Tootsville.UI.makeIDFromTitle is a function with lambda list: (title)

\*/

## 10.469 Tootsville.UI.makePrettyDialog

### 10.469.1 Function

Tootsville.UI.makePrettyDialog is a function with lambda list: (title,text,accept,cancel,resolve)■

Make a basic dialog box with a title, text, accept and cancel buttons. and call resolve function with user input later.



## **10.470 Tootsville.UI.makePrompt**

### **10.470.1 Function**

Tootsville.UI.makePrompt is a function with lambda list: (prompt, resolve)

WRITE ME

## **10.471 Tootsville.UI.onFirstClick**

### **10.471.1 Function**

Tootsville.UI.onFirstClick is nullary function.

WRITEME — this function is not yet documented.

## **10.472 Tootsville.UI.quit**

### **10.472.1 Function**

Tootsville.UI.quit is nullary function.

Request the user's confirmation to quit (or not)

## **10.473 Tootsville.UI.runCommand**

### **10.473.1 Function**

Tootsville.UI.runCommand is a function with lambda list: (command, event)

## **10.474 Tootsville.UI.say**

### **10.474.1 Function**

Tootsville.UI.say is a function with lambda list: (speech)

WRITEME

## **10.475 Tootsville.UI.setFullscreen**

### **10.475.1 Function**

Tootsville.UI.setFullscreen is a function with lambda list: (really)

Set fullscreen or windowed mode.

## **10.476 Tootsville.UI.setFullscreenFromNavigator**

### **10.476.1 Function**

Tootsville.UI.setFullscreenFromNavigator is nullary function.

Set the Fullscreen control panel toggle based on the current state of the navigator.

## **10.477 Tootsville.UI.signOut**

### **10.477.1 Function**

Tootsville.UI.signOut is nullary function.

Request the user's confirmation to quit (or not)



## **10.478 Tootsville.UI.slowLoadingWatchdog**

### **10.478.1 Function**

Tootsville.UI.slowLoadingWatchdog is nullary function.

Put up a warning about slow loading.

## 10.479 Tootsville.UI.takeOneStep

### 10.479.1 Function

Tootsville.UI.takeOneStep is a function with lambda list:  $(\delta x, \delta z)$

Walk one step in any direction.

For keyboard or gamepad inputs.

## **10.480 Tootsville.UI.toggleFullscreen**

### **10.480.1 Function**

Tootsville.UI.toggleFullscreen is nullary function.

Toggle between fullscreen and windowed mode

## 10.481 Tootsville.UI.useActiveItem

### 10.481.1 Function

Tootsville.UI.useActiveItem is a function with lambda list: (entity)

WRITEME — this function is not yet documented.

## 10.482 Tootsville.Util.assertValidHostName

### 10.482.1 Function

Tootsville.Util.assertValidHostName is a function with lambda list: (hostName)

Ensure that `hostName` is a valid hostname for the game cluster we're in.

## 10.483 Tootsville.Util.ensureServersReachable

### 10.483.1 Function

Tootsville.Util.ensureServersReachable is nullary function.

Check for the game REST server.

Calls <https://game.tootsville.org/meta-game/ping> and complains to the player if it can't be reached.

## **10.484 Tootsville.Util.equalP**

### **10.484.1 Function**

Tootsville.Util.equalP is a function with lambda list: (x, y)

Check for value equality of two objects

## 10.485 Tootsville.Util.infinityAwaits

### 10.485.1 Function

Tootsville.Util.infinityAwaits is a function with lambda list: (command, fromType, params)

Submit an Infinity Mode command to the servers or peers, but wait for the next reply of the given type for a Promised call-back. Note that the Gatekeeper still gets a chance to handle any returned values, this just registers a Promise to listen for the reply.



## **10.486 Tootsville.Util.loadScript**

### **10.486.1 Function**

Tootsville.Util.loadScript is a function with lambda list: (src)

Load the Javascript referenced by SRC into the page

## 10.487 Tootsville.Util.rest

### 10.487.1 Function

Tootsville.Util.rest is a function with lambda list: (method, uri, body, headers)

The main REST client.

<b>method</b>	GET, PUT, or POST
<b>uri</b>	The URI to access under the game host.
<b>body</b>	A JSON body for a PUT or POST
<b>headers</b>	An object which maps to additional headers to be set on the request. X-Infinity-Auth will be set when logged in; Accept and Content-Type will be defaulted to application/json if not set.

## 10.488 Tootsville.decodeTime

### 10.488.1 Function

Tootsville.decodeTime is nullary function.

Decode the current time as a Tootsville year, month, day, hour, &c.

The returned object has the following fields

- year
- month
- day (of month)
- hour
- min
- sec
- julian (day of year)
- weekday
- otherMonthDay
- pinkMonthDay

## 10.489 Tootsville.gamepadLayouts

### 10.489.1 Variable

The known layouts and mappings to names of several popular types of controllers.

This provides a name for the type of controller based upon its USB ID string (hex codes), as well as names for its buttons and axes. This will allow the end user to configure their device in a more user-friendly way; eg, by identifying a button as “A” rather than “button 0” they will be more easily able to identify the controls. (Seriously, who can remember if “Start” is button 8 or 7?)

Supported (so far) are all the gamepads I (BRP) use, which is to say:

- Generic NES-style USB gamepad
- Nintendo Switch USB gamepad
- SEGA Saturn style USB gamepad
- XBox 360 USB gamepad

## **10.490 Tootsville.host**

### **10.490.1 Variable**

The main container object under which nearly all other modules are located, to avoid potential namespace conflicts with other, loaded Javascript modules.

## **10.491 Tootsville.universalTimeOffset**

### **10.491.1 Variable**

The difference between Universal time and Unix time.

## **10.492 Tootsville.updateClock**

### **10.492.1 Function**

Tootsville.updateClock is nullary function.

Update the displayed clock on the screen.

## 10.493 `window.interpretTootColor`

### 10.493.1 Function

`window.interpretTootColor` is a function with lambda list: (name)

Translate the color named `name` into HTML-style hex code.

Strings which are not recognized as color names are expected to already be an HTML-style hex code.



## **10.494 window.onGoogleYoloLoad**

### **10.494.1 Function**

window.onGoogleYoloLoad is nullary function.



## 11 Credits

Tootsville is built upon a plethora of software. This is an attempt to convey at least a partial enumeration of the credits.

First, the most directly responsible:

Tootsville is a production of the Corporation for Inter-World Tourism and Adventuring, a not-for-profit corporation in the State of Florida, United States.

Tootsville V by Bruce-Robert Pocock at the Corporation for Inter-World Tourism and Adventuring.

Special thanks to Ali Dolan, Mariaelisa Greenwood, Richard Harnden, Levi Mc Call, and Zephyr Salz.

In memory of the contributions of Maureen Kenny (RIP).

Tootsville IV by Brandon Booker, Gene Cronk, Robert Dawson, Eric Feiling, Tim Hays, Sean King, Mark Mc Corkle, Cassandra Nichol, Bruce-Robert Pocock, and Ed Winkelman at Res Interactive, LLC.

### 11.1 Major Support Software

The following support software is used in the development of Tootsville:

- The Linux<sup>®</sup> Kernel
- The Gnu Operating System, by the Free Software Foundation
- The Fedora Distribution
- Emacs text editor and integrated development environment (IDE)
- Firefox web browser
- Chromium web browser
- Gimp graphics editor
- Inkscape graphics editor
- Blender graphics editor
- FFMPEG video and audio transcoder
- Audacity sound editor
- Rosegarden music editor

We'd also like to mention that we test with the following web browsers

- Firefox for macOS and Microsoft Windows
- Epiphany (Gnome Web)
- Google Chrome for Linux, macOS, and Microsoft Windows
- Opera
- Microsoft Edge for macOS and Microsoft Windows
- Apple Safari for macOS

## 11.2 Systems

The following systems (libraries) are compiled into the Tootsville server

### 11.2.1 System Tootsville

The server software monolith for REST services of Tootsville.org

Author: Bruce-Robert Pocock <BRPocock@ciwta.org>

License: AGPL v3+

### 11.2.2 System Twilio

Simple access to some of the Twilio API

Author: Bruce-Robert Pocock <BRPocock@ciwta.org>

License: AGPL v3+

### 11.2.3 System Thread-Pool-Taskmaster

Use a thread pool for a Taskmaster

Author: Bruce-Robert Pocock <brpocock@ciwta.org>

License: AGPL v3+

### 11.2.4 System Verbose

A logging framework using the piping library.

Author: Nicolas Hafner <shinmera@tymoon.eu>

License: zlib

### 11.2.5 System Documentation-Utills

A few simple tools to help you with documenting your library.

Author: Nicolas Hafner <shinmera@tymoon.eu>

Maintainer: Nicolas Hafner <shinmera@tymoon.eu>

License: zlib

### 11.2.6 System Trivial-Indent

A very simple library to allow indentation hints for SWANK.

Author: Nicolas Hafner <shinmera@tymoon.eu>

Maintainer: Nicolas Hafner <shinmera@tymoon.eu>

License: zlib

### 11.2.7 System Dissect

A lib for introspecting the call stack and active restarts.

Author: Nicolas Hafner <shinmera@tymoon.eu>

Maintainer: Nicolas Hafner <shinmera@tymoon.eu>

License: zlib

### 11.2.8 System Local-Time

A library for manipulating dates and times, based on a paper by Erik Naggum

Author: Daniel Lowe <dlowe@dlowe.net>

License: BSD

local-time Copyright (c) 2005-2012 by Daniel Lowe

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Thanks to those of you who have helped me make LOCAL-TIME an engaging and worthwhile project!

- \* Matthew Danish <mdanish@andrew.cmu.edu>
- \* The #lisp crew on irc.freenode.net
- \* Vladimir Sekissov <svg@surnet.ru>
- \* Attila Lendvai <attila.lendvai@gmail.com>
- \* Tomi Borbely <tomi.borbely@gmail.com>
- \* Denys Rtveliashvili <rtvd@mail.ru>
- \* Levente Meszaros <levente.meszaros@gmail.com>
- \* Arjan Wekking <arjan@streamtech.nl>
- \* Nikolai Matiushev <egao1980@gmail.com>
- \* Thomas Rake <zzzap1957@gmail.com>

---

Apologies to anyone I didn't mention (please let me know).

Daniel Lowe  
<dlowe@dlowe.net>

### 11.2.9 System Piping

A library to enable simple message pipelines.

Author: Nicolas Hafner <shinmera@tymoon.eu>

License: zlib

### 11.2.10 System Alexandria

Alexandria is a collection of portable public domain utilities.

Author: Nikodemus Siivola and others.

License: Public Domain / 0-clause MIT

### 11.2.11 System Rollbar

CL support for reporting to Rollbar

Author: Bruce-Robert Pocock

Maintainer: Bruce-Robert Pocock

License: BSD

### 11.2.12 System Oliphant

Various utilities that I use in different projects

Author: Bruce-Robert Fenn Pocock

Maintainer: Bruce-Robert Fenn Pocock

License: AGPLv3

### 11.2.13 System Usocket

Universal socket library for Common Lisp

Author: Erik Enge & Erik Huelsmann

Maintainer: Chun Tian (binghe) & Hans Huebner

License: MIT

### 11.2.14 System Sb-Bsd-Sockets

### 11.2.15 System Trivial-Gray-Streams

Compatibility layer for Gray Streams (see <http://www.cliki.net/Gray%20streams>).

Author: David Lichteblau

Maintainer: Anton Vodonosov <avodonosov@yandex.ru>

License: MIT

Copyright (c) 2005 David Lichteblau

Copyright (c) 2013 Anton Vodonosov <avodonosov@yandex.ru>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge,

publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 11.2.16 System Trivial-Garbage

Portable finalizers, weak hash-tables and weak pointers.

Author: Luis Oliveira <loliveira@common-lisp.net>

License: Public Domain

### 11.2.17 System St-Json

JSON in- and output

Author: Marijn Haverbeke <marijnh@gmail.com>

License: BSD

### 11.2.18 System Sqlite

CL-SQLITE package is an interface to the SQLite embedded relational database engine.

Author: Kalyanov Dmitry <Kalyanov.Dmitry@gmail.com>

Maintainer: Jacek Zlydach <cl-sqlite@jacek.zlydach.pl>

License: Public Domain

### 11.2.19 System Iterate

Jonathan Amsterdam's iterator/ gatherer/ accumulator facility

License: MIT

### 11.2.20 System Split-Sequence

Splits a sequence into a list of subsequences delimited by objects satisfying a test.

Author: Arthur Lemmens <alemmens@xs4all.nl>

Maintainer: Sharp Lispers <sharplispers@googlegroups.com>

License: MIT

### 11.2.21 System Prepl

### 11.2.22 System Named-Readtables

Library that creates a namespace for named readtable akin to the namespace of packages.

Author: Tobias C. Rittweiler <trittweiler@common-lisp.net>

Maintainer: Gábor Melis <mega@retes.hu>

License: BSD, see LICENSE

### 11.2.23 System Conium

### 11.2.24 System Closer-Mop

Closer to MOP is a compatibility layer that rectifies many of the absent or incorrect CLOS MOP features across a broad range of Common Lisp implementations.

Author: Pascal Costanza

License: MIT-style license

### 11.2.25 System Parse-Number

Number parsing library

Author: Matthew Danish <mrd@debian.org>

Maintainer: Sharp Lispers <sharplispers@googlegroups.com>

License: BSD 3-Clause

### 11.2.26 System Langutils

Language utilities

Author: Ian Eslick

License: BSD

### 11.2.27 System Stdutils

Standard Utilities

Author: Ian Eslick <eslick@media.mit.edu>

License: BSD

### 11.2.28 System S-Xml-Rpc

Common Lisp XML-RPC Package

Author: Sven Van Caekenberghe <svc@mac.com>

Maintainer: Sven Van Caekenberghe <>, Brian Mastenbrook <>, Rudi Schlatte <>, Pierre Neidhardt <mail@ambrevar.xyz>

License: Lesser Lisp General Public License (LLGPL)

### 11.2.29 System S-Xml

Simple Common Lisp XML Parser

Author: Sven Van Caekenberghe <svc@mac.com>

Maintainer: Sven Van Caekenberghe <svc@mac.com>, Brian Mastenbrook <>, Rudi Schlatte <>

License: Lisp Lesser General Public License (LLGPL)



### 11.2.30 System Cffi

The Common Foreign Function Interface

Author: James Bielman <jamesjb@jamesjb.com>

Maintainer: Luis Oliveira <loliveira@common-lisp.net>

License: MIT

### 11.2.31 System Babel

Babel, a charset conversion library.

Author: Luis Oliveira <loliveira@common-lisp.net>

License: MIT

### 11.2.32 System Trivial-Features

Ensures consistent \*FEATURES\* across multiple CLs.

Author: Luis Oliveira <loliveira@common-lisp.net>

License: MIT

### 11.2.33 System Cl-Unicode

Portable Unicode Library

License: BSD-2-Clause

### 11.2.34 System Cl-Unicode/ Base

License: BSD-2-Clause

### 11.2.35 System Cl-Readline

Common Lisp bindings to GNU Readline library

Author: Mark Karpov

License: GNU GPL, version 3

### 11.2.36 System Cl-Oauth

Common Lisp OAuth implementation

Maintainer: Leslie P. Polzer <polzer@gnu.org>

License: LLGPL

### 11.2.37 System Puri

Portable Universal Resource Identifier Library

Maintainer: Kevin M. Rosenberg <kmr@debian.org>

License: GNU Lesser General Public License

### 11.2.38 System F-Underscore

a tiny library of functional programming utils placed into the public domain.

the idea is to make functional programs shorter and easier to read without resorting to syntax [like arc's square bracket unary function syntax]

Author: Nick Allen <nallen05@gmail.com>

### 11.2.39 System Anaphora

The Anaphoric Macro Package from Hell

Author: Nikodemus Siivola <nikodemus@random-state.net>

License: Public Domain

### 11.2.40 System Ironclad

A cryptographic toolkit written in pure Common Lisp

Author: Nathan Froyd <froydnj@gmail.com>

Maintainer: Guillaume LE VAILLANT <glv@posteo.net>

License: BSD 3-Clause

### 11.2.41 System Sb-Posix

### 11.2.42 System Sb-Rotate-Byte

### 11.2.43 System Cl-Fad

Portable pathname library

License: BSD-2-Clause

### 11.2.44 System Buildapp

Buildapp is an application for SBCL and CCL that configures and saves an executable Common Lisp image or non-executable core.

Author: Zach Beane <xach@xach.com>

License: BSD

### 11.2.45 System Apply-Argv

Apply-argv is a library for parsing command line arguments.

Author: Peter von Etter

License: LLGPL

### 11.2.46 System Dreamhost

Access the Dreamhost API

Author: Bruce-Robert Pocock <brpocock@ciwta.org>

License: AGPL v3+

### 11.2.47 System Uuid

UUID Generation

Author: Boian Tzonev

Maintainer: Boian Tzonev

License: LLGPL

### 11.2.48 System Trivial-Utf-8

A small library for doing UTF-8-based input and output.

Author: Marijn Haverbeke <marijnh@gmail.com>

Maintainer: Gábor Melis <mega@retes.hu>

License: ZLIB

Copyright (c) Marijn Haverbeke

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

### 11.2.49 System Uiop

### 11.2.50 System Trivial-Signal

Unix signal handling library.

Author: Eitaro Fukamachi

License: Public Domain

### 11.2.51 System Trivial-Ldap

TRIVIAL-LDAP is a one file, all Common Lisp client implementation of parts of RFC 2261.

Author: Kevin Montuori

Maintainer: Raymond Wiker <rwiker@gmail.com>

License: Clarified Artistic License

### 11.2.52 System Yacc

A LALR(1) parser generator for Common Lisp

Author: Juliusz Chroboczek <jch@pps.jussieu.fr>

License: MIT/ X11

Copyright (c) 2005-2009 by Juliusz Chroboczek

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### **11.2.53 System Cl+Ssl**

Common Lisp interface to OpenSSL.

Author: Eric Marsden, Jochen Schmidt, David Lichteblau

License: MIT

### **11.2.54 System Flexi-Streams**

Flexible bivalent streams for Common Lisp

License: BSD-2-Clause

### **11.2.55 System Trivial-Backtrace**

trivial-backtrace

Author: Gary Warren King <gwking@metabang.com> and contributors

Maintainer: Gary Warren King <gwking@metabang.com> and contributors

License: MIT Style license

Copyright (c) 2008-2008 Gary Warren King (gwking@metabang.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2005-2007 Dr. Edi Weitz

BSD style license: <http://www.opensource.org/licenses/bsd-license.php>

### **11.2.56 System Symbol-Munger**

Functions to convert between the spacing and capitalization conventions of various environments

License: BSD

### **11.2.57 System Swank**

### **11.2.58 System Pngload**

A reader for the PNG image format.

Author: Michael Fiano <mail@mfiano.net>, Bart Botta <00003b@gmail.com>

License: MIT

### **11.2.59 System Zpb-Exif**

Read EXIF data from image files

Author: Zachary Beane <xach@xach.com>

License: BSD

### **11.2.60 System Swap-Bytes**

Optimized byte-swapping primitives.

Author: Stas Boukarev <stassats@gmail.com>

Maintainer: Stelian Ionescu <sionescu@cddr.org>

License: MIT

### **11.2.61 System Parse-Float**

Parse floating point values in strings.

Author: Sumant Oemrawsingh

License: Public Domain

### 11.2.62 System 3bz

deflate decompressor

Author: Bart Botta <00003b at gmail.com>

License: MIT

### 11.2.63 System Nibbles

A library for accessing octet-addressed blocks of data in big- and little-endian orders

Author: Nathan Froyd <froydnj@gmail.com>

Maintainer: Sharp Lispers <sharplispers@googlegroups.com>

License: BSD-style (<http://opensource.org/licenses/BSD-3-Clause>)

### 11.2.64 System Lparallel

Parallelism for Common Lisp

Author: James M. Lawrence <llmjml@gmail.com>

License: BSD

### 11.2.65 System Jonathan

High performance JSON encoder and decoder. Currently support: SBCL, CCL.

Author: Rudolph-Miller

License: MIT

### 11.2.66 System Cl-Annot

Python-like Annotation Syntax for Common Lisp

Author: Tomohiro Matsuyama

License: LLGPL

### 11.2.67 System Proc-Parse

Procedural vector parser

Author: Eitaro Fukamachi

License: BSD 2-Clause

### 11.2.68 System Sb-Cltl2

### 11.2.69 System Trivial-Types

Trivial type definitions

Author: Tomohiro Matsuyama

License: LLGPL

### 11.2.70 System Fast-Io

Alternative I/ O mechanism to a stream or vector

Author: Ryan Pavlik

License: MIT

### 11.2.71 System Static-Vectors

Create vectors allocated in static memory.

Author: Stelian Ionescu <sionescu@cddr.org>

License: MIT

### 11.2.72 System Cl-Syntax-Annot

CL-Syntax Reader Syntax for cl-annot

Author: Tomohiro Matsuyama

License: LLGPL

### 11.2.73 System Cl-Syntax

Reader Syntax Coventions for Common Lisp and SLIME

Author: Tomohiro Matsuyama

License: LLGPL

### 11.2.74 System Hunchensocket

WebSockets for Hunchentoot

Author: capitaomorte <<https://github.com/capitaomorte>>

License: MIT

Copyright (C) 2011 Alexander Kahl <[e-user@fsfe.org](mailto:e-user@fsfe.org)>

2014 João Távora

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR 'AS IS' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS

SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### 11.2.75 System Chunga

### 11.2.76 System Hunchentoot

Hunchentoot is a HTTP server based on USOCKET and BORDEAUX-THREADS. It supports HTTP 1.1, serves static files, has a simple framework for user-defined handlers and can be extended through subclassing.

License: BSD-2-Clause

### 11.2.77 System Rfc2388

Implementation of RFC 2388

Author: Janis Dzerins <jonis@latnet.lv>

License: Simplified BSD

### 11.2.78 System Md5

The MD5 Message-Digest Algorithm RFC 1321

Author: Pierre R. Mai <pmai@pmsf.de>

Maintainer: Pierre R. Mai <pmai@pmsf.de>

License: Public Domain

This software has been placed into the public domain.

This software is "as is", and has no warranty of any kind. The authors assume no responsibility for the consequences of any use of this software.

Additionally for all parts of this software ("The Work" below) authored by Pierre R. Mai copyright is waived under the CC0 as affirmed below:

Creative Commons Legal Code

CC0 1.0 Universal

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS DOCUMENT DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER.

Statement of Purpose



The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CCO with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CCO to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CCO on those rights.

1. Copyright and Related Rights. A Work made available under CCO may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

- i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;
- ii. moral rights retained by the original author(s) and/or performer(s);
- iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;
- iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;
- v. rights protecting the extraction, dissemination, use and reuse of data in a Work;
- vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and
- vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

- a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.
- b. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non

- infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.
- c. Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work or any use thereof, including without limitation any person's Copyright and Related Rights in the Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents, permissions or other rights required for any use of the Work.
  - d. Affirmer understands and acknowledges that Creative Commons is not a party to this document and has no duty or obligation with respect to this CCO or use of the Work.

### **11.2.79 System Fare-Memoization**

memoizing functions the correct, portable way

Author: Francois-Rene Rideau

License: MIT

### **11.2.80 System Envy**

Configuration switcher by an environment variable.

Author: Eitarow Fukamachi

License: BSD 2-Clause

### **11.2.81 System Drakma**

Full-featured http/ https client based on usocket

Author: Dr. Edi Weitz

License: BSD

### **11.2.82 System Chipz**

A library for decompressing deflate, zlib, and gzip data

Author: Nathan Froyd <froydnj@gmail.com>

Maintainer: Nathan Froyd <froydnj@gmail.com>

License: BSD style

### **11.2.83 System Dbd-Mysql**

Database driver for MySQL.

Author: Eitaro Fukamachi

License: LLGPL

### **11.2.84 System Cl-Mysql**

Common Lisp MySQL library bindings

Author: Steve Knight <stkni@yahoo.com>

Maintainer: Steve Knight <stkni@yahoo.com>

License: MIT

### 11.2.85 System Dbi

Database independent interface for Common Lisp

Author: Eitaro Fukamachi

License: LLGPL

### 11.2.86 System Darts.Lib.Email-Address

Parsing and formatting email addresses (RFC 5322 compliant)

Author: Dirk Esser

Maintainer: Dirk Eßer

License: MIT

### 11.2.87 System Cxml

Closure XML - a Common Lisp XML parser

Author: Gilbert Baumann, Henrik Motakef, David Lichteblau

Maintainer: Sharp Lispers <sharplispers@googlegroups.com>

License: LLGPL

Closure XML -- a Common Lisp XML parser

Copyright (c) 1999 by Gilbert Baumann

Copyright (c) 2003 by Henrik Motakef

Copyright (c) 2004 knowledgeTools Int. GmbH

Copyright (c) 2004,2005 David Lichteblau

Preamble to the Gnu Lesser General Public License

The concept of the GNU Lesser General Public License version 2.1 ("LGPL") has been adopted to govern the use and distribution of above-mentioned application. However, the LGPL uses terminology that is more appropriate for a program written in C than one written in Lisp. Nevertheless, the LGPL can still be applied to a Lisp program if certain clarifications are made. This document details those clarifications. Accordingly, the license for the open-source Lisp applications consists of this document plus the LGPL. Wherever there is a conflict between this document and the LGPL, this document takes precedence over the LGPL.

A "Library" in Lisp is a collection of Lisp functions, data and foreign modules. The form of the Library can be Lisp source code (for processing by an interpreter) or object code (usually the result of compilation of source code or built with some other mechanisms). Foreign modules are object code in a form that can be linked into a Lisp executable. When we

speak of functions we do so in the most general way to include, in addition, methods and unnamed functions. Lisp "data" is also a general term that includes the data structures resulting from defining Lisp classes. A Lisp application may include the same set of Lisp objects as does a Library, but this does not mean that the application is necessarily a "work based on the Library" it contains.

The Library consists of everything in the distribution file set before any modifications are made to the files. If any of the functions or classes in the Library are redefined in other files, then those redefinitions ARE considered a work based on the Library. If additional methods are added to generic functions in the Library, those additional methods are NOT considered a work based on the Library. If Library classes are subclassed, these subclasses are NOT considered a work based on the Library. If the Library is modified to explicitly call other functions that are neither part of Lisp itself nor an available add-on module to Lisp, then the functions called by the modified Library ARE considered a work based on the Library. The goal is to ensure that the Library will compile and run without getting undefined function errors.

It is permitted to add proprietary source code to the Library, but it must be done in a way such that the Library will still run without that proprietary code present. Section 5 of the LGPL distinguishes between the case of a library being dynamically linked at runtime and one being statically linked at build time. Section 5 of the LGPL states that the former results in an executable that is a "work that uses the Library." Section 5 of the LGPL states that the latter results in one that is a "derivative of the Library", which is therefore covered by the LGPL. Since Lisp only offers one choice, which is to link the Library into an executable at build time, we declare that, for the purpose applying the LGPL to the Library, an executable that results from linking a "work that uses the Library" with the Library is considered a "work that uses the Library" and is therefore NOT covered by the LGPL.

Because of this declaration, section 6 of LGPL is not applicable to the Library. However, in connection with each distribution of this executable, you must also deliver, in accordance with the terms and conditions of the LGPL, the source code of Library (or your derivative thereof) that is incorporated into this executable.

End of Document

---

GNU LESSER GENERAL PUBLIC LICENSE  
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that

there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of

free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from



such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If

identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to

distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference

directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot

use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent

infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by

the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

### 11.2.88 System Cxml/ Klacks

Closure XML -- a Common Lisp XML parser

Copyright (c) 1999 by Gilbert Baumann

Copyright (c) 2003 by Henrik Motakef  
Copyright (c) 2004 knowledgeTools Int. GmbH  
Copyright (c) 2004,2005 David Lichteblau

#### Preamble to the Gnu Lesser General Public License

The concept of the GNU Lesser General Public License version 2.1 ("LGPL") has been adopted to govern the use and distribution of above-mentioned application. However, the LGPL uses terminology that is more appropriate for a program written in C than one written in Lisp. Nevertheless, the LGPL can still be applied to a Lisp program if certain clarifications are made. This document details those clarifications. Accordingly, the license for the open-source Lisp applications consists of this document plus the LGPL. Wherever there is a conflict between this document and the LGPL, this document takes precedence over the LGPL.

A "Library" in Lisp is a collection of Lisp functions, data and foreign modules. The form of the Library can be Lisp source code (for processing by an interpreter) or object code (usually the result of compilation of source code or built with some other mechanisms). Foreign modules are object code in a form that can be linked into a Lisp executable. When we speak of functions we do so in the most general way to include, in addition, methods and unnamed functions. Lisp "data" is also a general term that includes the data structures resulting from defining Lisp classes. A Lisp application may include the same set of Lisp objects as does a Library, but this does not mean that the application is necessarily a "work based on the Library" it contains.

The Library consists of everything in the distribution file set before any modifications are made to the files. If any of the functions or classes in the Library are redefined in other files, then those redefinitions ARE considered a work based on the Library. If additional methods are added to generic functions in the Library, those additional methods are NOT considered a work based on the Library. If Library classes are subclassed, these subclasses are NOT considered a work based on the Library. If the Library is modified to explicitly call other functions that are neither part of Lisp itself nor an available add-on module to Lisp, then the functions called by the modified Library ARE considered a work based on the Library. The goal is to ensure that the Library will compile and run without getting undefined function errors.

It is permitted to add proprietary source code to the Library, but it must be done in a way such that the Library will still run without that proprietary code present. Section 5 of the LGPL distinguishes between the case of a library being dynamically linked at runtime and one being statically linked at build time. Section 5 of the LGPL states that the



former results in an executable that is a "work that uses the Library." Section 5 of the LGPL states that the latter results in one that is a "derivative of the Library", which is therefore covered by the LGPL. Since Lisp only offers one choice, which is to link the Library into an executable at build time, we declare that, for the purpose applying the LGPL to the Library, an executable that results from linking a "work that uses the Library" with the Library is considered a "work that uses the Library" and is therefore NOT covered by the LGPL.

Because of this declaration, section 6 of LGPL is not applicable to the Library. However, in connection with each distribution of this executable, you must also deliver, in accordance with the terms and conditions of the LGPL, the source code of Library (or your derivative thereof) that is incorporated into this executable.

End of Document

---

GNU LESSER GENERAL PUBLIC LICENSE  
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get

it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the

entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the

ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object

file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials

specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies,



or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot

impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

## 11.2.89 System Cxml/ Xml

Closure XML -- a Common Lisp XML parser

Copyright (c) 1999 by Gilbert Baumann  
Copyright (c) 2003 by Henrik Motakef  
Copyright (c) 2004 knowledgeTools Int. GmbH  
Copyright (c) 2004,2005 David Lichteblau

Preamble to the Gnu Lesser General Public License

The concept of the GNU Lesser General Public License version 2.1 ("LGPL") has been adopted to govern the use and distribution of above-mentioned application. However, the LGPL uses terminology that is more appropriate for a program written in C than one written in Lisp. Nevertheless, the LGPL can still be applied to a Lisp program if certain clarifications are made. This document details those clarifications. Accordingly, the license for the open-source Lisp applications consists of this document plus the LGPL. Wherever there is a conflict between this document and the LGPL, this document takes precedence over the LGPL.

A "Library" in Lisp is a collection of Lisp functions, data and foreign modules. The form of the Library can be Lisp source code (for processing by an interpreter) or object code (usually the result of compilation of source code or built with some other mechanisms). Foreign modules are object code in a form that can be linked into a Lisp executable. When we speak of functions we do so in the most general way to include, in addition, methods and unnamed functions. Lisp "data" is also a general term that includes the data structures resulting from defining Lisp

classes. A Lisp application may include the same set of Lisp objects as does a Library, but this does not mean that the application is necessarily a "work based on the Library" it contains.

The Library consists of everything in the distribution file set before any modifications are made to the files. If any of the functions or classes in the Library are redefined in other files, then those redefinitions ARE considered a work based on the Library. If additional methods are added to generic functions in the Library, those additional methods are NOT considered a work based on the Library. If Library classes are subclassed, these subclasses are NOT considered a work based on the Library. If the Library is modified to explicitly call other functions that are neither part of Lisp itself nor an available add-on module to Lisp, then the functions called by the modified Library ARE considered a work based on the Library. The goal is to ensure that the Library will compile and run without getting undefined function errors.

It is permitted to add proprietary source code to the Library, but it must be done in a way such that the Library will still run without that proprietary code present. Section 5 of the LGPL distinguishes between the case of a library being dynamically linked at runtime and one being statically linked at build time. Section 5 of the LGPL states that the former results in an executable that is a "work that uses the Library." Section 5 of the LGPL states that the latter results in one that is a "derivative of the Library", which is therefore covered by the LGPL. Since Lisp only offers one choice, which is to link the Library into an executable at build time, we declare that, for the purpose applying the LGPL to the Library, an executable that results from linking a "work that uses the Library" with the Library is considered a "work that uses the Library" and is therefore NOT covered by the LGPL.

Because of this declaration, section 6 of LGPL is not applicable to the Library. However, in connection with each distribution of this executable, you must also deliver, in accordance with the terms and conditions of the LGPL, the source code of Library (or your derivative thereof) that is incorporated into this executable.

End of Document

---

GNU LESSER GENERAL PUBLIC LICENSE  
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original

author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating

system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does

and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those



sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not

excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

## 11.2.90 System Closure-Common

### 11.2.91 System Cxml/ Dom

Closure XML -- a Common Lisp XML parser

Copyright (c) 1999 by Gilbert Baumann  
Copyright (c) 2003 by Henrik Motakef  
Copyright (c) 2004 knowledgeTools Int. GmbH

Copyright (c) 2004,2005 David Lichteblau

#### Preamble to the Gnu Lesser General Public License

The concept of the GNU Lesser General Public License version 2.1 ("LGPL") has been adopted to govern the use and distribution of above-mentioned application. However, the LGPL uses terminology that is more appropriate for a program written in C than one written in Lisp. Nevertheless, the LGPL can still be applied to a Lisp program if certain clarifications are made. This document details those clarifications. Accordingly, the license for the open-source Lisp applications consists of this document plus the LGPL. Wherever there is a conflict between this document and the LGPL, this document takes precedence over the LGPL.

A "Library" in Lisp is a collection of Lisp functions, data and foreign modules. The form of the Library can be Lisp source code (for processing by an interpreter) or object code (usually the result of compilation of source code or built with some other mechanisms). Foreign modules are object code in a form that can be linked into a Lisp executable. When we speak of functions we do so in the most general way to include, in addition, methods and unnamed functions. Lisp "data" is also a general term that includes the data structures resulting from defining Lisp classes. A Lisp application may include the same set of Lisp objects as does a Library, but this does not mean that the application is necessarily a "work based on the Library" it contains.

The Library consists of everything in the distribution file set before any modifications are made to the files. If any of the functions or classes in the Library are redefined in other files, then those redefinitions ARE considered a work based on the Library. If additional methods are added to generic functions in the Library, those additional methods are NOT considered a work based on the Library. If Library classes are subclassed, these subclasses are NOT considered a work based on the Library. If the Library is modified to explicitly call other functions that are neither part of Lisp itself nor an available add-on module to Lisp, then the functions called by the modified Library ARE considered a work based on the Library. The goal is to ensure that the Library will compile and run without getting undefined function errors.

It is permitted to add proprietary source code to the Library, but it must be done in a way such that the Library will still run without that proprietary code present. Section 5 of the LGPL distinguishes between the case of a library being dynamically linked at runtime and one being statically linked at build time. Section 5 of the LGPL states that the former results in an executable that is a "work that uses the Library." Section 5 of the LGPL states that the latter results in one that is a

"derivative of the Library", which is therefore covered by the LGPL. Since Lisp only offers one choice, which is to link the Library into an executable at build time, we declare that, for the purpose applying the LGPL to the Library, an executable that results from linking a "work that uses the Library" with the Library is considered a "work that uses the Library" and is therefore NOT covered by the LGPL.

Because of this declaration, section 6 of LGPL is not applicable to the Library. However, in connection with each distribution of this executable, you must also deliver, in accordance with the terms and conditions of the LGPL, the source code of Library (or your derivative thereof) that is incorporated into this executable.

End of Document

---

GNU LESSER GENERAL PUBLIC LICENSE  
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do



these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with

the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data

prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in

these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the

Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.



This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN

WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

### 11.2.92 System Cloudb

### 11.2.93 System S-Base64

Common Lisp Base64 Package

Author: Sven Van Caekenberghe <svc@mac.com>

Maintainer: Sven Van Caekenberghe <svc@mac.com>

License: Lesser Lisp General Public License (LLGPL)

### 11.2.94 System Parensript

Lisp to JavaScript transpiler

Author: Manuel Odendahl <manuel@bl0rg.net>

Maintainer: Vladimir Sedach <vas@oneofus.la>

License: BSD-3-Clause

Copyright (c) 2005 Manuel Odendahl <manuel@bl0rg.net>

Copyright (c) 2005-2006 Edward Marco Baringer <mb@bese.it>

Copyright (c) 2007-2013, 2018 Vladimir Sedach <vas@oneofus.la>

Copyright (c) 2008, 2009 Travis Cross <tc@travislists.com>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its

contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### 11.2.95 System Cljwt-Custom

JSON Web Token library

Author: Grim Schjetne <grim@schjetne.se>

License: LGPLv3+

### 11.2.96 System Yason

JSON parser/ encoder

Author: Hans Huebner <hans@huebner.org>

License: BSD

### 11.2.97 System Cl-Smtp

Common Lisp smtp client.

Author: Jan Idzikowski <jidzikowski@common-lisp.net>

Maintainer: Jan Idzikowski <jidzikowski@common-lisp.net>

License: LLGPL

### 11.2.98 System Cl-Ppcre

Perl-compatible regular expression library

Author: Dr. Edi Weitz

License: BSD

### 11.2.99 System Cl-Memcached

Fast, thread-safe library to interface with the Memcached Object Cache.

Author: quasi <quasi@quasilabs.in>

License: MIT

### 11.2.100 System Pooler

Generic thread-safe pooling facility for your library.

Author: quasi <quasi@quasilabs.in>

License: MIT

### 11.2.101 System Sb-Concurrency

### 11.2.102 System Cl-Dbi

Author: Eitaro Fukamachi

License: LLGPL

### 11.2.103 System Cl-Base64

Base64 encoding and decoding with URI support.

Author: Kevin M. Rosenberg based on initial code by Juri Pakaste

Maintainer: Kevin M. Rosenberg <kmr@debian.org>

License: BSD-style

Copyright (c) 2002-2003 by Kevin Rosenberg

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the Authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### 11.2.104 System Bordeaux-Threads

Bordeaux Threads makes writing portable multi-threaded apps simple.

Author: Stelian Ionescu <sionescu@cddr.org>

License: MIT

### 11.2.105 System Global-Vars

Define efficient global variables.

Author: James M. Lawrence <llmjml@gmail.com>

License: MIT

## 11.3 The Steel Bank Common Lisp compiler

Tootsville is developed and compiled using the Steel Bank Common Lisp compiler.

```
 -*- coding: utf-8; mode: text; -*-
```

```

    The programmers of old were mysterious and profound. We
    cannot fathom their thoughts, so all we do is describe their
    appearance.
```

```

    Aware, like a fox crossing the water. Alert, like a general
    on the battlefield. Kind, like a hostess greeting her guests.
```

```

    Simple, like uncarved blocks of wood. Opaque, like black
    pools in darkened caves.
```

```

    Who can tell the secrets of their hearts and minds?
```

```

    The answer exists only in the Tao.
```

```

    -- Geoffrey James, "The Tao of Programming"
```

### BROAD OUTLINE

SBCL is derived from the 18b version of CMU CL.

Most of CMU CL was originally written as part of the CMU Common Lisp project at Carnegie Mellon University. According to the documentation in CMU CL 18b,

```

    Organizationally, CMU Common Lisp was a small, mostly autonomous
    part within the Mach operating system project. The CMU CL project
    was more of a tool development effort than a research project.
```

```

    The project started out as Spice Lisp, which provided a modern
    Lisp implementation for use in the CMU community.
```

and

```

    CMU CL has been under continuous development since the early 1980's
    (concurrent with the Common Lisp standardization effort.)
```

Apparently most of the CMU Common Lisp implementors moved on to work on the Gwydion environment for Dylan.

CMU CL's CLOS implementation is derived from the PCL reference

implementation written at Xerox PARC.

CMU CL's implementation of the LOOP macro was derived from code from Symbolics, which was derived from code from MIT.

CMU CL had many individual author credits in the source files. In the sometimes-extensive rearrangements which were required to make SBCL bootstrap itself cleanly, it was tedious to try keep such credits attached to individual source files, so they have been moved here instead.

Bill Newman <william.newman@airmail.net> did this transformation, and so any errors made are probably his. Corrections would be appreciated.

#### MORE DETAILS ON SBCL'S CLOS CODE

The original headers of the PCL files contained the following text:

```
;;; Any person obtaining a copy of this software is requested to send their
;;; name and post office or electronic mail address to:
;;;   CommonLoops Coordinator
;;;   Xerox PARC
;;;   3333 Coyote Hill Rd.
;;;   Palo Alto, CA 94304
;;; (or send Arpanet mail to CommonLoops-Coordinator.pa@Xerox.arpa)
;;;
;;; Suggestions, comments and requests for improvements are also welcome.
```

This was intended for the original incarnation of the PCL code as a portable reference implementation. Since our version of the code has had its portability hacked out of it, it's no longer particularly relevant to any coordinated PCL effort (which probably doesn't exist any more anyway). Therefore, this contact information has been deleted from the PCL file headers.

A few files in the original CMU CL 18b src/pcl/ directory did not carry such Xerox copyright notices:

- \* Some code was originally written by Douglas T. Crosher for CMU CL:
  - \*\* the Gray streams implementation
  - \*\* the implementation of DOCUMENTATION as methods of a generic function
- \* generic-functions.lisp seems to have been machine-generated.

The comments in the CMU CL 18b version of the PCL code walker, src/pcl/walk.lisp, said in part

```
;;; a simple code walker, based IN PART on: (roll the credits)
```

```

;;;    Larry Masinter's Masterscope
;;;    Moon's Common Lisp code walker
;;;    Gary Drescher's code walker
;;;    Larry Masinter's simple code walker
;;;    .
;;;    .
;;;    boy, thats fair (I hope).

```

#### MORE DETAILS ON SBCL'S LOOP CODE

The src/code/loop.lisp file from CMU CL 18b had the following credits-related information in it:

```

;;; The LOOP iteration macro is one of a number of pieces of code
;;; originally developed at MIT for which free distribution has been
;;; permitted, as long as the code is not sold for profit, and as long
;;; as notification of MIT's interest in the code is preserved.
;;;
;;; This version of LOOP, which is almost entirely rewritten both as
;;; clean-up and to conform with the ANSI Lisp LOOP standard, started
;;; life as MIT LOOP version 829 (which was a part of NIL, possibly
;;; never released).
;;;
;;; A "light revision" was performed by me (Glenn Burke) while at
;;; Palladian Software in April 1986, to make the code run in Common
;;; Lisp. This revision was informally distributed to a number of
;;; people, and was sort of the "MIT" version of LOOP for running in
;;; Common Lisp.
;;;
;;; A later more drastic revision was performed at Palladian perhaps a
;;; year later. This version was more thoroughly Common Lisp in style,
;;; with a few miscellaneous internal improvements and extensions. I
;;; have lost track of this source, apparently never having moved it to
;;; the MIT distribution point. I do not remember if it was ever
;;; distributed.
;;;
;;; The revision for the ANSI standard is based on the code of my April
;;; 1986 version, with almost everything redesigned and/or rewritten.

```

The date of the M.I.T. copyright statement falls around the time described in these comments. The dates on the Symbolics copyright statement are all later -- the earliest is 1989.

#### MORE DETAILS ON OTHER SBCL CODE FROM CMU CL

CMU CL's symbol (but not package) code (code/symbol.lisp) was originally written by Scott Fahlman and updated and maintained by Skef Wholey.

The CMU CL reader (code/reader.lisp) was originally the Spice Lisp reader, written by David Dill and with support for packages added by Lee Schumacher. David Dill also wrote the sharpmacro support (code/sharpm.lisp).

CMU CL's package code was rewritten by Rob MacLachlan based on an earlier version by Lee Schumacher. It also includes DEFPACKAGE by Dan Zigmond, and WITH-PACKAGE-ITERATOR written by Blaine Burks. William Lott also rewrote the DEFPACKAGE and DO-FOO-SYMBOLS stuff.

CMU CL's string code (code/string.lisp) was originally written by David Dill, then rewritten by Skef Wholey, Bill Chiles, and Rob MacLachlan.

Various code in the system originated with "Spice Lisp", which was apparently a predecessor to the CMU CL project. Much of that was originally written by Skef Wholey:

- code/seq.lisp, generic sequence functions, and COERCE
- code/array.lisp, general array stuff
- SXHASH
- code/list.lisp, list functions (based on code from Joe Ginder and Carl Ebeling)

The CMU CL seq.lisp code also gave credits for later work by Jim Muller and Bill Chiles.

The modules system (code/module.lisp, containing REQUIRE, PROVIDE, and friends, now deprecated by ANSI) was written by Jim Muller and rewritten by Bill Chiles.

The CMU CL garbage collector was credited to "Christopher Hoover, Rob MacLachlan, Dave McDonald, et al." in the CMU CL code/gc.lisp file, with some extra code for the MIPS port credited to Christopher Hoover alone. The credits on the original "gc.c", "Stop and Copy GC based on Cheney's algorithm", said "written by Christopher Hoover".

Guy Steele wrote the original character functions  
code/char.lisp

They were subsequently rewritten by David Dill, speeded up by Scott Fahlman, and rewritten without fonts and with a new type system by Rob MachLachlan.

Lee Schumacher made the Spice Lisp version of backquote. The comment in the CMU CL sources suggests he based it on someone else's code for



some other Lisp system, but doesn't say which. A note in the CMU CL code to pretty-print backquote expressions says that unparsing support was provided by Miles Bader.

The CMU implementations of the Common Lisp query functions Y-OR-N-P and YES-OR-NO-P were originally written by Walter van Roggen, and updated and modified by Rob MacLachlan and Bill Chiles.

The CMU CL sort functions (code/sort.lisp) were written by Jim Large, hacked on and maintained by Skef Wholey, and rewritten by Bill Chiles.

Most of the internals of the Python compiler seem to have been originally written by Robert MacLachlan:

- the type system and associated "cold load hack magic"
  - code/typedefs.lisp
  - code/class.lisp
  - code/type-init.lisp
  - etc.
- the lexical environment database
  - compiler/globaldb.lisp, etc.
- the IR1 representation and optimizer
  - compiler/ir1\*.lisp, etc.
- the IR2 representation and optimizer
  - compiler/ir2\*.lisp, etc.
- many concrete optimizations
  - compiler/srctran.lisp (with some code adapted from CLC by Wholey and Fahlman)
  - compiler/float-tran.lisp, etc.
- information about optimization of known functions
  - compiler/fndb.lisp
- debug information representation
  - compiler/debug.lisp, compiler/debug-dump.lisp
- memory pools to reduce consing by reusing compiler objects
  - compiler/alloc.lisp
- toplevel interface functions and drivers
  - compiler/main.lisp

Besides writing the compiler, and various other work mentioned elsewhere, Robert MacLachlan was also credited with tuning the implementation of streams for Unix files, and writing

- various floating point support code
  - code/float-trap.lisp, floating point traps
  - code/float.lisp, misc. support a la INTEGER-DECODE-FLOAT
- low-level time functions
  - code/time.lisp

William Lott is also credited with writing or heavily maintaining some parts of the CMU CL compiler. He was responsible for lifting

compiler/meta-vmdef.lisp out of compiler/vmdef.lisp, and also wrote  
 various optimizations  
     compiler/array-tran.lisp  
     compiler/saptran.lisp  
     compiler/seqtran.lisp (with some code adapted from an older  
         seqtran written by Wholey and Fahlman)  
 the separable compiler backend  
     compiler/backend.lisp  
     compiler/generic/utils.lisp  
 the implementation of LOAD-TIME-VALUE  
     compiler/ltv.lisp  
 the most recent version of the assembler  
     compiler/new-assem.lisp  
 vop statistics gathering  
     compiler/statcount.lisp  
 centralized information about machine-dependent and..  
 ..machine-independent FOO, with  
     compiler/generic/vm-fndb.lisp, FOO=function signatures  
     compiler/generic/vm-typertran.lisp, FOO=type ops  
     compiler/generic/objdef.lisp, FOO=object representation  
     compiler/generic/printtype.lisp, FOO=primitive types

Also, Christopher Hoover and William Lott wrote compiler/generic/vm-macs.lisp to centralize information about machine-dependent macros and constants.

Sean Hallgren is credited with most of the Alpha backend. Julian Dolby created the CMU CL Alpha/Linux port. Douglas Crosher added complex-float support.

The original PPC backend was the work of Gary Byers. Some bug fixes and other changes to update it for current CMUCL interfaces were made by Eric Marsden and Douglas Crosher

The CMU CL machine-independent disassembler (compiler/disassem.lisp) was written by Miles Bader.

Parts of the CMU CL system were credited to Skef Wholey and Rob MacLachlan jointly, perhaps because they were originally part of Spice Lisp and were then heavily modified:

code/load.lisp, the loader, including all the FASL stuff  
 code/macros.lisp, various fundamental macros  
 code/mipsstrops.lisp, primitives for hacking strings  
 code/purify.lisp, implementation of PURIFY  
 code/stream.lisp, stream functions  
 code/lispinit.lisp, cold startup  
 code/profile.lisp, the profiler

Bill Chiles also modified code/macros.lisp. Much of the implementation

of PURIFY was rewritten in C by William Lott.

The CMU CL number functions (code/number.lisp) were written by Rob MacLachlan, but acknowledge much code "derived from code written by William Lott, Dave McDonald, Jim Large, Scott Fahlman, etc."

CMU CL's weak pointer support (code/weak.lisp) was written by Christopher Hoover.

The CMU CL DEFSTRUCT system was credited to Rob MacLachlan, William Lott and Skef Wholey jointly.

The FDEFINITION system for handling arbitrary function names (a la (SETF FOO)) was originally written by Rob MacLachlan. It was modified by Bill Chiles to add encapsulation, and modified more by William Lott to add FDEFN objects.

The CMU CL condition system (code/error.lisp) was based on some prototyping code written by Kent Pitman at Symbolics.

The CMU CL HASH-TABLE system was originally written by Skef Wholey for Spice Lisp, then rewritten by William Lott, then rewritten again by Douglas T. Crosher.

The support code for environment queries (a la LONG-SITE-NAME), the DOCUMENTATION function, and the DRIBBLE function was written and maintained "mostly by Skef Wholey and Rob MacLachlan. Scott Fahlman, Dan Aronson, and Steve Handerson did stuff here too." The same credit statement was given for the original Mach OS interface code.

The CMU CL printer, print.lisp, was credited as "written by Neal Feinberg, Bill Maddox, Steven Handerson, and Skef Wholey, and modified by various CMU Common Lisp maintainers." The comments on the float printer said specifically that it was written by Bill Maddox. The comments on bignum printing said specifically that it was written by Steven Handerson (based on Skef's idea), and that it was rewritten by William Lott to remove assumptions about length of fixnums on the MIPS port.

The comments in the main body of the CMU CL debugger  
code/debug.lisp  
say that it was written by Bill Chiles. Some other related files  
code/debug-int.lisp, programmer's interface to the debugger  
code/ntrace.lisp, tracing facility based on breakpoints  
say they were written by Bill Chiles and Rob MacLachlan.  
The related file  
src/debug-vm.lisp, low-level support for :FUNCTION-END breakpoints

was written by William Lott.

The CMU CL GENESIS cold load system, compiler/generic/new-genesis.lisp, was originally written by Skef Wholey, then jazzed up for packages by Rob MacLachlan, then completely rewritten by William Lott for the MIPS port.

The CMU CL IR1 interpreter was written by Bill Chiles and Robert MacLachlan.

Various CMU CL support code was written by William Lott:

- the bytecode interpreter
  - code/byte-interp.lisp
- bitblt-ish operations a la SYSTEM-AREA-COPY
  - code/bit-bash.lisp
- Unix interface
  - code/fd-stream.lisp, Unix file descriptors as Lisp streams
  - code/filesys.lisp, other Unix filesystem interface stuff
- handling errors signalled from assembly code
  - code/interr.lisp
  - compiler/generic/interr.lisp
- finalization based on weak pointers
  - code/final.lisp
- irrational numeric functions
  - code/irrat.lisp
- the pretty printer
  - code/pprint.lisp
- predicates (both type predicates and EQUAL and friends)
  - code/pred.lisp
- saving the current Lisp image as a core file
  - code/save.lisp
- handling Unix signals
  - code/signal.lisp
- implementing FORMAT
  - code/format.lisp

The ALIEN facility seems to have been written largely by Rob MacLachlan and William Lott. The CMU CL comments say "rewritten again, this time by William Lott and Rob MacLachlan," but don't identify who else might have been involved in earlier versions.

The comments in CMU CL's code/final.lisp say "the idea really was Chris Hoover's". The comments in CMU CL's code/pprint.lisp say "Algorithm stolen from Richard Waters' XP." The comments in CMU CL's code/format.lisp say "with lots of stuff stolen from the previous version by David Adam and later rewritten by Bill Maddox."

Jim Muller was credited with fixing seq.lisp.

CMU CL's time printing logic, in code/format-time.lisp, was written by Jim Healy.

Bill Chiles was credited with fixing/updating seq.lisp after Jim Muller.

The CMU CL machine/filesystem-independent pathname functions (code/pathname.lisp) were written by William Lott, Paul Gleichauf, and Rob MacLachlan, based on an earlier version written by Jim Large and Rob MacLachlan.

Besides writing the original versions of the things credited to him above, William Lott rewrote, updated, and cleaned up various stuff:

```
code/array.lisp
code/serve-event.lisp
```

The INSPECT function was originally written by Blaine Burks.

The CMU CL DESCRIBE facility was originally written by "Skef Wholey or Rob MacLachlan", according to the comments in the CMU CL sources. It was cleaned up and reorganized by Blaine Burks, then ported and cleaned up more by Rob MacLachlan. Also, since the split from CMU CL, the SBCL DESCRIBE facility was rewritten as a generic function and so become entangled with some DESCRIBE code which was distributed as part of PCL.

The implementation of the Mersenne Twister RNG used in SBCL is based on an implementation written by Douglas T. Crosher and Raymond Toy, which was placed in the public domain with permission from M. Matsumoto.

Comments in the CMU CL version of FreeBSD-os.c said it came from an OSF version by Sean Hallgren, later hacked by Paul Werkowski, with generational conservative GC support added by Douglas Crosher.

Comments in the CMU CL version of linux-os.c said it came from the FreeBSD-os.c version, morfed to Linux by Peter Van Eynde in July 1996.

Comments in the CMU CL version of backtrace.c said it was "originally from Rob's version" (presumably Robert Maclachlan).

Comments in the CMU CL version of purify.c said it had stack direction changes, x86/CGC stack scavenging, and static blue bag stuff (all for x86 port?) by Paul Werkowski, 1995, 1996; and bug fixes, x86 code movement support, and x86/gencgc stack scavenging by Douglas Crosher, 1996, 1997, 1998.

According to comments in the source files, much of the CMU CL version of the x86 support code

```

assembly/x86/alloc.lisp
assembly/x86/arith.lisp
assembly/x86/array.lisp
assembly/x86/assem-rtns.lisp
compiler/x86/alloc.lisp
compiler/x86/arith.lisp
compiler/x86/c-call.lisp
compiler/x86/call.lisp
compiler/x86/cell.lisp
compiler/x86/char.lisp
compiler/x86/debug.lisp
compiler/x86/float.lisp
compiler/x86/insts.lisp
compiler/x86/macros.lisp
compiler/x86/memory.lisp
compiler/x86/move.lisp
compiler/x86/nlx.lisp
compiler/x86/parms.lisp
compiler/x86/pred.lisp
compiler/x86/print.lisp
compiler/x86/sap.lisp
compiler/x86/static-fn.lisp
compiler/x86/subprim.lisp
compiler/x86/system.lisp
compiler/x86/type-vops.lisp
compiler/x86/values.lisp
compiler/x86/vm.lisp

```

was originally written by William Lott, then debugged by Paul Werkowski, and in some cases later enhanced and further debugged by Douglas T. Crosher; and the x86 runtime support code,

```
x86-assem.S
```

was written by Paul F. Werkowski and Douglas T. Crosher.

The CMU CL user manual (doc/cmu-user/cmu-user.tex) says that the X86 FreeBSD port was originally contributed by Paul Werkowski, and Peter VanEynde took the FreeBSD port and created a Linux version.

According to comments in src/code/bsd-os.lisp, work on the generic BSD port was done by Skef Wholey, Rob MacLachlan, Scott Fahlman, Dan Aronson, and Steve Handerson.

Douglas Crosher wrote code to support Gray streams, added X86 support for the debugger and relocatable code, wrote a conservative generational GC for the X86 port. He also added X86-specific

extensions to support stack groups and multiprocessing, but these are not present in SBCL

The CMU CL user manual credits Robert MacLachlan as editor. A chapter on the CMU CL interprocess communication extensions (not supported in SBCL) was contributed by William Lott and Bill Chiles.

Peter VanEynde also contributed a variety of `#+HIGH-SECURITY` patches to CMU CL, to provide additional safety, especially through runtime checking on various tricky cases of standard functions (e.g. MAP with complicated result types, and interactions of various variants of STREAM).

Raymond Toy wrote CMU CL's PROPAGATE-FLOAT-TYPE extension and various other floating point optimizations. (In SBCL, the PROPAGATE-FLOAT-TYPE entry in `*FEATURES*` first became SB-PROPAGATE-FLOAT-TYPE, then went away completely as the code became an unconditional part of the system.)

CMU CL's long float support was written by Douglas T. Crosher.

Paul Werkowski turned the Mach OS support code into Linux OS support code.

Versions of the RUN-PROGRAM extension were written first by David McDonald, then by Jim Healy and Bill Chiles, then by William Lott.

#### MORE DETAILS ON THE TRANSITION FROM CMU CL

Bill Newman did the original conversion from CMU CL 18b to a form which could bootstrap itself cleanly, on Linux/x86 only. Although they may not have realized it at the time, Rob MacLachlan and Peter Van Eynde were very helpful, RAM by posting a clear explanation of what GENESIS is supposed to be doing and PVE by maintaining a version of CMU CL which worked on Debian, so that I had something to refer to whenever I got stuck.

#### CREDITS SINCE THE RELEASE OF SBCL

(Note: (1) This is probably incomplete, since there's no systematic procedure for updating it. (2) Some more details are available in the NEWS file, in the project's CVS change logs, and in the archives of the sbcl-devel mailing list. (3) In this, as in other parts of SBCL, patches are welcome. Don't be shy.)

Martin Atzmueller:

He reported many bugs, fixed many bugs, ported various fixes from CMU CL, and helped clean up various stale bug data. (He has been unusually energetic at this. As of sbcl-0.6.9.10, the total number of bugs involved likely exceeded 100. Since then, I've lost count. See the CVS logs.)

Daniel Barlow:

His contributions have included support for shared object loading (from CMUCL), the Cheney GC for non-x86 ports (from CMUCL), Alpha and PPC ports (from CMUCL), control stack exhaustion checking (new), native threads support for x86 Linux (new), and the initial x86-64 backend (new). He also refactored the garbage collectors for understandability, wrote code (e.g. grovel-headers.c and stat\_wrapper stuff) to find machine-dependent and OS-dependent constants automatically, and was original author of the asdf, asdf-install, sb-bsd-sockets, sb-executable, sb-grovel and sb-posix contrib packages.

Zach Beane:

He provided a number of additions to SB-POSIX, implemented the original timer facility on which SBCL's timers are based. and also contributed the :SAVE-RUNTIME-OPTIONS support for SAVE-LISP-AND-DIE.

James Bielman:

He assisted in work on the port to the Windows operating system, and was instrumental in :EXECUTABLE support for SAVE-LISP-AND-DIE.

Alastair Bridgewater:

He contributed a port of the system to the Windows operating system.

Robert E. Brown:

He has reported various bugs and submitted several patches, especially improving removing gratuitous efficiencies in the standard library.

Cadabra, Inc. (later merged into GoTo.com):

They hired Bill Newman to do some consulting for them, including the implementation of EQUALP hash tables for CMU CL; then agreed to release the EQUALP code into the public domain, giving SBCL (and CMU CL) its EQUALP hash tables.

Douglas Crosher:

He continued to improve CMU CL after SBCL forked from it, creating many patches which were directly applicable to SBCL. Notable examples include fixes for various compiler bugs, the implementation of CL:DEFINE-SYMBOL-MACRO, and a generalization of the type system's handling of the CONS type to allow ANSI-style (CONS FOO BAR) types.



Larry D'Anna:

He provided several parts of SB-CLTL2 environment access, and has also worked on bugs in the IR2 conversion stage of the compiler.

Alexey Dejneka:

He fixed many, many bugs on various themes, and has done a tremendous amount of work on the compiler in particular, fixing bugs and refactoring.

Paul Dietz:

He is in the process of writing a comprehensive test suite for the requirements of the ANSI Common Lisp standard. Already, at the halfway stage, it has caught hundreds of bugs in SBCL, and provided simple test cases for them. His random crash tester has caught an old deep problem in the implementation of the stack analysis phase in the compiler.

Brian Downing:

He fixed the linker problems for building SBCL on Mac OS X. He found and fixed the cause of backtraces failing for undefined functions and assembly routines. He wrote the core of SBCL's alternative interpreter-based EVAL.

Miles Egan:

He creates binary packages of SBCL releases for Red Hat and other (which?) platforms.

Helmut Eller:

A lot of the code in the SB-INTROSPECT and SB-COVER contrib modules was originally written by him for Slime/Swank.

Lutz Euler:

He made a large number of improvements to the x86-64 disassembler.

Andreas Fuchs:

He provides infrastructure for monitoring build and performance regressions of SBCL. He assisted with the integration of the Unicode work.

Stephan Frank:

He contributed the SB-GMP contrib to exploit libgmp in bignum and ratio arithmetic.

Nathan Froyd:

He has fixed various bugs, and also done a lot of internal cleanup, not visible at the user level but important for

maintenance. (E.g. converting the PCL code to use LOOP instead of the old weird pre-ANSI ITERATE macro so that the code can be read without being an expert in ancient languages and so that we can delete a thousand lines of implement-ITERATE macrology from the codebase.)

Bruno Haible:

He devised an accurate continued-fraction-based implementation of RATIONALIZE, replacing a less-accurate version inherited from primordial CMUCL.

Cyrus Harmon:

He fixed many PPC FFI and callback bugs. He ported Raymond Toy's work on the generational garbage collector for PPC to Linux, finding and fixing other SBCL bugs in the process.

Matthias Hoelzl:

He reported and fixed COMPILER's misbehavior on macros.

Daisuke Homma:

He added support for SunOS on x86 processors.

ITA Software:

They hired Juho Snellman as a consultant to work on improvements to SBCL, to be released into the public domain. The work they've funded includes faster compilation, various improvements to the statistical profiler, the SB-COVER code coverage tool, the interpreter-based evaluator and the IR2-based single-stepper.

Espen S Johnsen:

He provided an ANSI-compliant version of CHANGE-CLASS for PCL.

Teemu Kalvas:

He worked on Unicode support for SBCL, including parsing the Unicode character database, restoring the FAST-READ-CHAR optimization and developing external format support.

Dmitry Kalyanov:

His work was crucial in bringing the Windows backend forward; he implemented pthreads and ported SB-THREAD to this platform.

Yaroslav Kavenchuk:

He implemented several missing features and fixed many bugs in the win32 port. He also worked on external-format support for SB-ALIEN.

Anton Kovalenko:

He introduced a safepoint-based stop-the-world protocol and greatly contributed to features and bugfixes related to the Windows port.

Richard M Kreyter:

He added documentation support for CLOS slot readers and writers, provided several SB-POSIX and NetBSD patches, and cleaned up several of the filesystem/pathname interfaces.

Frederik Kuivinen:

He showed how to implement the DEBUG-RETURN functionality.

Arthur Lemmens:

He found and fixed a number of SBCL bugs while partially porting SBCL to bootstrap under Lispworks for Windows.

David Lichteblau:

He repeatedly failed to update his entry in this file.

Robert MacLachlan:

He has continued to answer questions about, and contribute fixes to, the CMU CL project. Some of these fixes, especially for compiler problems, has been invaluable to the CMU CL project and, by porting, invaluable to the SBCL project as well.

Pierre Mai:

He has continued to work on CMU CL since the SBCL fork, and also patched code to SBCL to enable dynamic loading of object files under OpenBSD. He contributed to the port of SBCL to MacOS X, implementing the Lisp side of the PowerOpen ABI.

Eric Marsden:

Some of his fixes to CMU CL since the SBCL fork have been ported to SBCL. He also maintains the cl-benchmark package, which gives us some idea of how our performance changes compared to earlier releases and to other implementations. He assisted in development of Unicode support for SBCL.

Antonio Martinez-Shotton:

He has contributed a number of bug fixes and bug reports to SBCL.

Brian Mastenbrook:

He contributed to and extensively maintained the port of SBCL to MacOS X. His contributions include overcoming binary compatibility issues between different versions of dlcompat on Darwin, other linker fixes, and signal handler bugfixes.

Dave McDonald:

He made a lot of progress toward getting SBCL to be bootstrappable under CLISP.

Gabor Melis:

He mainly worked on robustness related to signal handling, threads, timers with small excursions to constraint propagation, weak hash tables (based on CMUCL code) and optimizing x86/x86-64 calling convention.

Perry E. Metzger:

He ported SBCL to NetBSD with newer signals, building on the work of Valtteri Vuorikoski. He also provided various cleanups to the C runtime.

Gerd Moellman:

He has made many cleanups and improvements, small and large, in CMU CL (mostly in PCL), which we have gratefully ported to SBCL. Of particular note is his ctor MAKE-INSTANCE optimization, which is both faster in the typical case than the old optimizations in PCL and less buggy.

Timothy Moore:

He designed and implemented the original CMUCL linkage-table, on which the SBCL implementation thereof is based.

William ("Bill") Newman:

He continued to maintain SBCL after the fork, increasing ANSI compliance, fixing bugs, regularizing the internals of the system, deleting unused extensions, improving performance in some areas (especially sequence functions and non-simple vectors), updating documentation, and even, for better or worse, getting rid of various functionality (e.g. the byte interpreter).

NIIMI Satoshi:

He contributed a number of fixes to the FreeBSD port, implemented some external-formats and JOIN-THREAD, and also worked on the :EXECUTABLE support.

Patrik Nordebo:

He contributed to the port of SBCL to MacOS X, finding solutions for ABI and assembly syntax differences between Darwin and Linux.

Luís Oliveira:

He contributed to the port of SBCL to the Windows operating system, particularly in the area of FFI.

Scott Parish:

He ported SBCL to OpenBSD-with-ELF.

Timothy Ritchey:

He implemented SB-BSD-SOCKETS support for the win32 port.

Tobias Rittweiler

He has made several contributions relating to source locations, pretty printing, SB-INTROSPECT, and the reader.

Kevin M. Rosenberg:

He provided the ACL-style toplevel (sb-aclrepl contrib module), and a number of MOP-related bug reports. He also creates the official Debian packages of SBCL.

Joshua Ross:

He fixed some bugs relating to foreign calls and callbacks on the Linux PowerPC platform.

Christophe Rhodes:

He ported SBCL to SPARC (based on the CMUCL backend), made various port-related and SPARC-related changes (like \*BACKEND-SUBFEATURES\*), made many fixes and improvements in the compiler's type system, has essentially completed the work to enable bootstrapping SBCL under unrelated (non-SBCL, non-CMU-CL) Common Lisps. He participated in the modernization of SBCL's CLOS implementation, implemented the treatment of compiler notes as restartable conditions, provided optimizations to compiler output, and contributed in other ways as well.

Stig Erik Sandø:

He showed how to convince the GNU toolchain to build SBCL in a way which supports callbacks from C code into SBCL.

Rudi Schlatte:

He ported Paul Foley's simple-streams implementation from cmucl, converted the sbcl manual to Texinfo and wrote a documentation string extractor that keeps function documentation in the manual current.

Thiemo Seufer:

He modernized the MIPS backend, fixing many bugs, and assisted in cleaning up the C runtime code.

Julian Squires:

He worked on Unicode support for the PowerPC platform.

Nikodemus Siivola:

He provided build fixes, in particular to tame the SunOS toolchain, implemented package locks, ported the linkage-table code from CMUCL, reimplemented STEP, implemented the compare-and-swap interface, and has fixed many bugs besides.

Juho Snellman:

He provided a number of bug fixes and performance enhancements to the compiler, the standard library functions, and to the garbage collector. He ported and enhanced the statistical profiler written by Gerd Moellmann for CMU CL. He completed the work on the x86-64 port of SBCL.

Brian Spilsbury:

He wrote Unicode-capable versions of SBCL's character, string, and stream types and operations on them. (These versions did not end up in the system, but did to a large extent influence the support which finally did get merged.)

Robert Swindells:

He ported SBCL to NetBSD/Sparc.

Raymond Toy:

He continued to work on CMU CL after the SBCL fork, especially on floating point stuff. Various patches and fixes of his have been ported to SBCL, including his Sparc port of linkage-table.

Larry Valkama:

He resurrected the HPUX port, and worked on the HPPA backend in general.

Peter Van Eynde:

He wrestled the CLISP test suite into a mostly portable test suite (clocc ansi-test) which can be used on SBCL, provided a slew of bug reports resulting from that, and submitted many other bug reports as well.

Valtteri Vuorikoski:

He ported SBCL to NetBSD, and also fixed a long-standing bug in DEFSTRUCT with respect to colliding accessor names.

Colin Walters:

His  $O(N)$  implementation of the general case of MAP, posted on the cmucl-imp@cons.org mailing list, was the inspiration for similar MAP code added in sbcl-0.6.8.

Cheuksan Edward Wang:

He assisted in debugging the SBCL x86-64 backend.

Raymond Wiker:

He ported sbcl-0.6.3 back to FreeBSD, restoring the ancestral CMU CL support for FreeBSD and updating it for the changes made from FreeBSD version 3 to FreeBSD version 4. He also ported the CMU CL extension RUN-PROGRAM, and related code, to SBCL.

INITIALS GLOSSARY (helpful when reading comments, CVS commit logs, etc.)

AB Alastair Bridgewater  
AK Anton Kovalenko  
AL Arthur Lemmens  
APD Alexey Dejneka  
CLH Cyrus Harmon  
CSR Christophe Rhodes  
DB Daniel Barlow (also "dan")  
DFL David Lichteblau  
DTC Douglas Crosher  
JES Juho Snellman  
JRXR Joshua Ross  
LAV Larry Valkama  
LEU Lutz Euler  
MG Gabor Melis  
MNA Martin Atzmueller  
NJF Nathan Froyd  
NS Nikodemus Siivola  
PFD Paul F. Dietz  
PRM Pierre Mai  
PVE Peter Van Eynde  
PK/PVK Paul-Virak Khuong  
PW Paul Werkowski  
RAM Robert MacLachlan  
RLT Raymond Toy  
TCR Tobias Rittweiler  
THS Thiemo Seufer  
VJA Vincent Arkesteijn  
WHN William ("Bill") Newman

## 11.4 Javascript Tools

The Javascript front-end to Tootsville is developed using the following additional tools:

- Babylon.JS 3D rendering library
- Google Closure Javascript compressor





## 12 Conclusion

### 12.1 License

Tootsville V and the Romance II game system are based upon Braque, Copyright © 2006, 2007, Bruce-Robert Pocock, and Appius Claudius Caecus and the Romance Game System version I, Copyright © 2008-2015 Bruce-Robert Pocock.

Tootsville V is Copyright © 2008-2017 Bruce-Robert Pocock; © 2018-2021 The Corporation for Inter-World Tourism and Adventuring.

Tootsville and the The Toots characters were created by Res Interactive, LLC, a dissolved Florida limited-liability corporation.

#### 12.1.1 Tootsville Contents

Tootsville original Toot (UltraToot) character design, logos, design, layout, world design, and other contents are Copyright © 2016-2017, Bruce-Robert Pocock, and Copyright © 2018-2021 The Corporation for Inter-World Tourism and Adventuring. ALL RIGHTS RESERVED.

*The use of original Tootsville content in any work, in any medium, must be approved by CIWTA.*

If you would like to use Tootsville Materials of any kind in your own work, contact the Cadre at [cadre@ciwta.org](mailto:cadre@ciwta.org), or write to:

PO Box 23095  
Oakland Park, FL 33307-3095  
USA

#### 12.1.2 Additional Media Content

Tootsville incorporates a variety of Additional Media Content, in the form of 2D and 3D graphics, music, sound effects, movies, books, and other contents. Credit through the User Interface is given whenever practical. Here, too, is a partial listing of acknowledgements for Additional Media Content:

WRITEME

### 12.2 AGPL v3 License

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License (following) for more details.

#### 12.2.1 GNU AFFERO GENERAL PUBLIC LICENSE

Version 3, 19 November 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The GNU Affero General Public License is a free, copyleft license for software and other kinds of works, specifically designed to ensure cooperation with the community in the case of network server software.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, our General Public Licenses are intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

Developers that use our General Public Licenses protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License which gives you legal permission to copy, distribute and/or modify the software.

A secondary benefit of defending all users' freedom is that improvements made in alternate versions of the program, if they receive widespread use, become available for other developers to incorporate. Many developers of free software are heartened and encouraged by the resulting cooperation. However, in the case of software used on network servers, this result may fail to come about. The GNU General Public License permits making a modified version and letting the public access it on a server without ever releasing its source code to the public.

The GNU Affero General Public License is designed specifically to ensure that, in such cases, the modified source code becomes available to the community. It requires the operator of a network server to provide the source code of the modified version running there to the users of that server. Therefore, public use of a modified version, on a publicly accessible server, gives the public access to the source code of the modified version.

An older license, called the Affero General Public License and published by Affero, was designed to accomplish similar goals. This is a different license, not a version of the Affero GPL, but Affero has released a new version of the Affero GPL which permits relicensing under this license.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU Affero General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

## 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## **2. Basic Permissions.**

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

## **3. Protecting Users' Legal Rights From Anti-Circumvention Law.**

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

## **4. Conveying Verbatim Copies.**

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

## **5. Conveying Modified Source Versions.**

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

## 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and non-commercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third

party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## **8. Termination.**

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

## **9. Acceptance Not Required for Having Copies.**

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

## **10. Automatic Licensing of Downstream Recipients.**

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.



## 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

## **12. No Surrender of Others' Freedom.**

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

## **13. Remote Network Interaction; Use with the GNU General Public License.**

Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software. This Corresponding Source shall include the Corresponding Source for any work covered by version 3 of the GNU General Public License that is incorporated pursuant to the following paragraph.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the work with which it is combined will remain governed by version 3 of the GNU General Public License.

## **14. Revised Versions of this License.**

The Free Software Foundation may publish revised and/or new versions of the GNU Affero General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU Affero General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU Affero General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU Affero General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

### **15. Disclaimer of Warranty.**

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

### **16. Limitation of Liability.**

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### **17. Interpretation of Sections 15 and 16.**

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

## **How to Apply These Terms to Your New Programs**

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as published
by the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
```

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License along with this program. If not, see [<http://www.gnu.org/licenses/>](http://www.gnu.org/licenses/).

Also add information on how to contact you by electronic and paper mail.

If your software can interact with users remotely through a computer network, you should also make sure that it provides a way for users to get its source. For example, if your program is a web application, its interface could display a "Source" link that leads users to an archive of the code. There are many ways you could offer source, and different solutions will be better for different programs; see section 13 for the specific requirements.

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU AGPL, see [<http://www.gnu.org/licenses/>](http://www.gnu.org/licenses/).

# Appendix A Indices

## A.1 Concepts

### A

Authentication, Infinity Mode ..... 4  
 Authentication, REST ..... 4

### B

Bowling ..... 870

### C

Card Games ..... 864  
 Chœrogyllum Holidays ..... 26  
 Changes from 1.0 to 1.1 . . . 947, 972, 983, 1003, 1535  
 Changes from 1.0 to 1.2 ..... 990  
 Changes from 1.1 to 1.2 . . . 938, 942, 958, 971, 977,  
 983, 1535  
 Changes from 1.2 to 2.0 . . . 130, 131, 138, 142, 157,  
 158, 161, 182, 183, 184, 190, 191, 192, 193, 200, 202,  
 213, 218, 220, 224, 230, 233, 940, 942, 943, 946, 948,  
 951, 958, 963, 964, 967, 968, 970, 972, 975, 976, 977,  
 978, 979, 980, 982, 983, 985, 990, 1004, 1010, 1011,  
 1012, 1015, 1018, 1019, 1022, 1025, 1035, 1536, 1619  
 Client Tilde Commands ..... 1976  
 Clusters, types of ..... 4

### E

Emotes ..... 1017

### I

Infinity Mode Protocol ..... 592  
 Infinity Mode, Overview ..... 4

### K

Known bugs ..... 948

### M

Movement, Overview ..... 6

### P

Places, within the game ..... 971

### Q

Quaestor Events ..... 1020  
 Quiescing and Burgeoning ..... 994

### R

REST interface ..... 4  
 Romance Game System, what is ..... 3  
 Room Variables ..... 969

### S

Server Prompts and Replies ..... 991  
 Speaking, Speech ..... 1016  
 Speech, Overview ..... 6  
 SportsBall ..... 871

### T

Tootsville, what is ..... 3

### W

Walking the Line ..... 1028



**3**

3-Days-Ago ..... 364

**A**

Accept-Type-Equal ..... 366  
 Acceptor-Status-Message ..... 367  
 Accepts-Content-Type-P ..... 368  
 Active-Player ..... 369  
 Add-Charset ..... 370  
 Add-Contact ..... 371  
 Add-Or-Replace-Endpoint ..... 372  
 Addevent ..... 132  
 Admin-Message ..... 373  
 After-Slash ..... 374  
 Agent ..... 133  
 All-Connected ..... 375  
 All-Credits ..... 376  
 All-Links-To-Same-Person-P ..... 377  
 All-Symbols-Alphabetically ..... 378  
 Allowed-Base-Colors-Under-Pattern ..... 379  
 Allowed-Pattern-Colors-On-Base ..... 380  
 Altitude ..... 381  
 Answered-Child-Requests-By-Toot ..... 382  
 Apply-Config ..... 383  
 Apply-Extension-To-Template ..... 384  
 Arrange-Columns+Values-For-Find ..... 385  
 As-Response ..... 1764  
 Askme ..... 134  
 Assert-My-Character ..... 386  
 Associate-Credentials ..... 387  
 At ..... 135  
 Atom-Or-Comma-List ..... 388  
 Avatar-Avatar-Scale-X ..... 390  
 Avatar-Avatar-Scale-X, SetF ..... 390  
 Avatar-Avatar-Scale-Y ..... 391  
 Avatar-Avatar-Scale-Y, SetF ..... 391  
 Avatar-Avatar-Scale-Z ..... 392  
 Avatar-Avatar-Scale-Z, SetF ..... 392  
 Avatar-Has-Slot-P ..... 393  
 Avatar-Id ..... 394  
 Avatar-Id, SetF ..... 394  
 Avatar-Moniker ..... 395  
 Avatar-Moniker, SetF ..... 395  
 Avatar-P ..... 396  
 Avatar-Slot-Avatar ..... 398  
 Avatar-Slot-Avatar, SetF ..... 398  
 Avatar-Slot-Id ..... 399  
 Avatar-Slot-Id, SetF ..... 399  
 Avatar-Slot-P ..... 400  
 Avatar-Slot-Slot ..... 401  
 Avatar-Slot-Slot, SetF ..... 401  
 Avatar-Slot-Valence ..... 402  
 Avatar-Slot-Valence, SetF ..... 402  
 Average ..... 403  
 Ayt-Idle-Users ..... 404

**B**

Background-Gc ..... 405  
 Backtrace-Frame-To-Plist ..... 66  
 Bad-Request-Thing ..... 407  
 Bad-Request-Thing, SetF ..... 407  
 Ban ..... 136  
 Banhammer ..... 137  
 Banhammer-Ip-Address ..... 408  
 Banner ..... 409  
 Banner/ Error-Output ..... 410  
 Banner/ Log ..... 411  
 Banner/ Query-Id ..... 412  
 Banner/ Standard-Output ..... 413  
 Banner/ Trace-Output ..... 414  
 Base64-From-Uri-Form ..... 415  
 Base64-To-Uuid ..... 416  
 Beam ..... 138  
 Before-Save-Normalize ..... 418  
 Bool-Sort ..... 419  
 Broadcast ..... 420  
 Build-Simple-Column-Query ..... 421  
 Build-Simple-Query ..... 422  
 Builder-Toot-P ..... 423  
 Surgeon-Quiesced-State ..... 424  
 Byte-Vector-To-Integer ..... 425  
 Bytes-Json ..... 426

**C**

Cal-Month ..... 14  
 Cal-Month-Header ..... 15  
 Cal-Month-Header.Html ..... 16  
 Cal-Month.Html ..... 17  
 Cal-Month/ Print-Holiday-Footnotes ..... 18  
 Cal-Year ..... 19  
 Call-Infinity-From-Rest ..... 427  
 Call-Infinity-From-Stream ..... 428  
 Cassandra-Add-To-Blacklist ..... 429  
 Cassandra-Add-To-Redlist ..... 430  
 Cassandra-Boot ..... 431  
 Cassandra-Filter ..... 432  
 Cassandra-Obnoxious-Filter ..... 433  
 Cassandra-Remove-From-Blacklist ..... 434  
 Cassandra-Remove-From-Redlist ..... 435  
 Census ..... 139  
 Chain-Debugger-Hook ..... 67  
 Character-Music-Music ..... 438  
 Character-Music-Music, SetF ..... 438  
 Character-Music-P ..... 439  
 Character-Music-Toot ..... 440  
 Character-Music-Toot, SetF ..... 440  
 Chdir ..... 441  
 Check-Alexa ..... 442  
 Check-Alexa-Signature ..... 443  
 Check-Alexa-Signature-Cert-Chain-Url ..... 444  
 Check-Alexa-Timestamp-Tolerance ..... 445  
 Check-Arg-Type ..... 446  
 Check-Cert-Chain-Valid ..... 447

Check-Cert-Dates-Valid	448	Concat	499
Check-Firebase-Id-Token	449	Condition-Name	500
Check-Pattern-On-Base-Color	450	Condition-Slots	501
Check-Toot-Name	451	Condition-Telemetry	69
Check-X.509-San	452	Config	502
Child-Request-Allowed-At	455	Configure	70
Child-Request-Allowed-At, SetF	455	Connect-Cache	503
Child-Request-Allowed-For	456	Connect-Databases	504
Child-Request-Allowed-For, SetF	456	Connect-Maria	505
Child-Request-Allowed-Until	457	Connect-Time	506
Child-Request-Denied-At	458	Connected-Toot-Names	507
Child-Request-Denied-At, SetF	458	Connected-Toots	508
Child-Request-P	459	Consider-Child-Kick	509
Child-Request-Placed-At	460	Constituent-Char-P	71
Child-Request-Placed-At, SetF	460	Constituentp	510
Child-Request-Response	461	Contact-Added	512
Child-Request-Response, SetF	461	Contact-Added, SetF	512
Child-Request-Toot	462	Contact-Contact	513
Child-Request-Toot, SetF	462	Contact-Contact, SetF	513
Child-Request-Uuid	463	Contact-Last-Used	514
Child-Request-Uuid, SetF	463	Contact-Last-Used, SetF	514
Classify-Error-Level	68	Contact-Owner	515
Clean-Ice-Credentials	464	Contact-Owner, SetF	515
Clean-Symbols	465	Contact-P	516
Clear-All-Endpoints	466	Contact-Starredp	517
Clearbadge	140	Contact-Starredp, SetF	517
Clearcache	141	Contact-Uuid	518
Clearevent	142	Contact-Uuid, SetF	518
Clearvar	143	Contents-To-Bytes	519
Cloneroom	144	Copy-Avatar	520
Clouds	467	Copy-Avatar-Slot	521
Cluster	468	Copy-Character-Music	522
Cluster-Name	469	Copy-Child-Request	523
Cluster-Net-Name	470	Copy-Color24	524
Color24-Blue	478	Copy-Contact	525
Color24-Blue, SetF	478	Copy-Credential	526
Color24-Green	479	Copy-Game-Point	527
Color24-Green, SetF	479	Copy-Gossip-Initiation	528
Color24-Hsv	480	Copy-Inventory-Item	529
Color24-Hue	481	Copy-Item	530
Color24-Name	482	Copy-Item-Template	531
Color24-P	483	Copy-Locale-Music	532
Color24-Red	484	Copy-Login	533
Color24-Red, SetF	484	Copy-Lot	534
Color24-Rgb	485	Copy-Metronome-Task	535
Color24-Saturation	486	Copy-Mist	536
Color24-To-Integer	487	Copy-Music	537
Color24-Value	488	Copy-Parent-Child	538
Color24/ =	489	Copy-Pattern	539
Color24=	490	Copy-Person	540
Column-Load-Mapping	491	Copy-Person-Link	541
Column-Load-Value	492	Copy-Place	542
Column-Normalizer	493	Copy-Quaestor-Event	543
Column-Save-Mapping	494	Copy-Sms	544
Column-Save-Value	495	Copy-Store-Item	545
Compute-Fountain-Peanuts-For-Score	496	Copy-Tcp-Client	546
Compute-Fountain-Random-Fairy-Dust	497	Copy-Terrain-Edge-Horz	547
Compute-Next-Keys-Update	498	Copy-Terrain-Edge-Vert	548





Dreamhost-Error-Details.....	41
Dress.....	150
Drop.....	151
Drop-Item.....	646
Dropkick.....	152
Dump-Credits.....	647
Dump-Global-Heightmap.....	648
Dumpthreads.....	153
<b>E</b>	
Email-Lhs.....	649
Enable-Sbcl-Ldb.....	650
Enable-Ssl-P.....	651
Enablepathfinder.....	154
Encode*-Universal-Time.....	23
Encode-Endpoint-Reply.....	652
Endpoint->Html.....	654
Endpoint->Openapi.....	655
Endpoint->Plist.....	656
Endpoint-Close.....	657
Endpoint-Close-Key.....	658
Endpoint-Content-Type.....	659
Endpoint-Delete-/ Users/ Me/ Toots/ Toot-Name->Json.....	660
Endpoint-Function.....	661
Endpoint-Get-/ ↳Html.....	700
Endpoint-Get-/ Favicon->Gif.....	663
Endpoint-Get-/ Favicon->Png.....	664
Endpoint-Get-/ Favicon/ Ico->Vnd.Microsoft.Icon.....	662
Endpoint-Get-/ Gossip/ Answers/ Uuid->Sdp.....	665
Endpoint-Get-/ Gossip/ Ice-Servers->Json.....	666
Endpoint-Get-/ Gossip/ Offers->Json.....	667
Endpoint-Get-/ Maintenance/ ↳Txt.....	668
Endpoint-Get-/ Meta-Game/ Headers->Json..	669
Endpoint-Get-/ Meta-Game/ Ping->Txt.....	670
Endpoint-Get-/ Meta-Game/ Services->Html.....	672
Endpoint-Get-/ Meta-Game/ Services->Json.....	673
Endpoint-Get-/ Meta-Game/ Services/ Old->Json.....	671
Endpoint-Get-/ Toots/ Toot-Name->Json.....	674
Endpoint-Get-/ Toots/ Toot-Name->Txt.....	675
Endpoint-Get-/ Users/ Me->Json.....	680
Endpoint-Get-/ Users/ Me->Txt.....	681
Endpoint-Get-/ Users/ Me/ Toots->Json.....	678
Endpoint-Get-/ Users/ Me/ Toots->Txt.....	679
Endpoint-Get-/ Users/ Me/ Toots/ Toot-Name->Json.....	676
Endpoint-Get-/ Users/ Me/ Toots/ Toot-Name->Txt.....	677
Endpoint-Get-/ Version/ About->Json.....	685
Endpoint-Get-/ Version/ About->Txt.....	686
Endpoint-Get-/ Version/ About/ Detail/ Param->Json.....	682
Endpoint-Get-/ Version/ About/ Detail/ Param->Txt.....	683
Endpoint-Get-/ World->Json.....	699
Endpoint-Get-/ World/ Clock/ Calendar/ Now/ Fragment->Html.....	687
Endpoint-Get-/ World/ Clock/ Calendar/ Year/ Year/ Fragment->Html.....	688
Endpoint-Get-/ World/ Clock/ Calendar/ Year/ Year/ Month/ Month->Html.....	690
Endpoint-Get-/ World/ Clock/ Calendar/ Year/ Year/ Month/ Month/ Fragment->Html.....	689
Endpoint-Get-/ World/ Clock/ Date->Txt....	693
Endpoint-Get-/ World/ Clock/ Date/ Abbrev->Txt.....	691
Endpoint-Get-/ World/ Clock/ Date/ Long->Txt.....	692
Endpoint-Get-/ World/ Clock/ Time->Json...	695
Endpoint-Get-/ World/ Clock/ Time->Txt....	696
Endpoint-Get-/ World/ Clock/ Time/ Detailed->Txt.....	694
Endpoint-Get-/ World/ Sky/ Tootanga/ Latitude/ Longitude->Json.....	697
Endpoint-Get-/ World/ Tootanga/ Latitude/ Longitude/ Altitude->Json.....	698
Endpoint-Hash.....	701
Endpoint-Kinda-Key.....	702
Endpoint-Method.....	703
Endpoint-Patch-/ Users/ Me->Json.....	704
Endpoint-Post-/ Gossip/ Alexa/ Chat/ Region/ Region->Json.....	705
Endpoint-Post-/ Gossip/ Alexa/ Clock/ Region/ Region->Json.....	706
Endpoint-Post-/ Gossip/ Alexa/ Info/ Region/ Region->Json.....	707
Endpoint-Post-/ Gossip/ Answers/ Uuid->Sdp.....	708
Endpoint-Post-/ Gossip/ Offers->Sdp.....	709
Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Call->Xml.....	710
Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Fax->Xml.....	711
Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Sms->Xml.....	712
Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Verify->Xml.....	713
Endpoint-Post-/ Gossip/ Twilio/ Incoming/ Whatsapp->Xml.....	714
Endpoint-Post-/ Login/ Child->Json.....	715
Endpoint-Post-/ Maintenance/ Buildapp->Nil.....	717
Endpoint-Post-/ Maintenance/ Buildapp/ Status->Nil.....	716
Endpoint-Post-/ Maintenance/ Hot-Reload->Nil.....	718
Endpoint-Post-/ Maintenance/ Quicklisp-Update->Nil.....	719

Endpoint-Post-/ Maintenance/ Quit→Nil ...	720	Endpoint-Post-/ World/ Infinity/ Get-Session-Apple→Json .....	749
Endpoint-Post-/ Maintenance/ Reload-Jscl→Nil .....	721	Endpoint-Post-/ World/ Infinity/ Get-Store-Item-Info→Json .....	750
Endpoint-Post-/ Toots→Json .....	722	Endpoint-Post-/ World/ Infinity/ Get-User-Lists→Json .....	751
Endpoint-Post-/ Users/ Me/ Play-With/ Toot-Name→Json .....	723	Endpoint-Post-/ World/ Infinity/ Get-Wallet→Json .....	752
Endpoint-Post-/ World/ Infinity→Json .....	787	Endpoint-Post-/ World/ Infinity/ Get-Zone-List→Json .....	753
Endpoint-Post-/ World/ Infinity/ Add-Furniture→Json .....	724	Endpoint-Post-/ World/ Infinity/ Give→Json .....	754
Endpoint-Post-/ World/ Infinity/ Add-Journal-Entry→Json .....	725	Endpoint-Post-/ World/ Infinity/ Go→Json .....	755
Endpoint-Post-/ World/ Infinity/ Add-To-List→Json .....	726	Endpoint-Post-/ World/ Infinity/ Init-User-Room→Json .....	756
Endpoint-Post-/ World/ Infinity/ Click→Json .....	727	Endpoint-Post-/ World/ Infinity/ Join→Json .....	757
Endpoint-Post-/ World/ Infinity/ Consider-Child-Approval→Json .....	728	Endpoint-Post-/ World/ Infinity/ Logout→Json .....	758
Endpoint-Post-/ World/ Infinity/ Create-User-House→Json .....	729	Endpoint-Post-/ World/ Infinity/ Mail-Customer-Service→Json .....	759
Endpoint-Post-/ World/ Infinity/ Delete-Mail-Message→Json .....	730	Endpoint-Post-/ World/ Infinity/ Peek-At-Inventory→Json .....	760
Endpoint-Post-/ World/ Infinity/ Doff→Json .....	732	Endpoint-Post-/ World/ Infinity/ Ping→Json .....	761
Endpoint-Post-/ World/ Infinity/ Dofff→Json .....	731	Endpoint-Post-/ World/ Infinity/ Play-With→Json .....	762
Endpoint-Post-/ World/ Infinity/ Don→Json .....	733	Endpoint-Post-/ World/ Infinity/ Prompt-Reply→Json .....	763
Endpoint-Post-/ World/ Infinity/ Echo→Json .....	734	Endpoint-Post-/ World/ Infinity/ Quiesce→Json .....	764
Endpoint-Post-/ World/ Infinity/ End-Event→Json .....	735	Endpoint-Post-/ World/ Infinity/ Read-Map→Json .....	765
Endpoint-Post-/ World/ Infinity/ Enumerate-Wear-Slots→Json .....	736	Endpoint-Post-/ World/ Infinity/ Remove-From-List→Json .....	766
Endpoint-Post-/ World/ Infinity/ Finger→Json .....	737	Endpoint-Post-/ World/ Infinity/ Report-Bug→Json .....	767
Endpoint-Post-/ World/ Infinity/ Game-Action→Json .....	738	Endpoint-Post-/ World/ Infinity/ Report-User→Json .....	768
Endpoint-Post-/ World/ Infinity/ Get-Avatars→Json .....	739	Endpoint-Post-/ World/ Infinity/ Request-Buddy→Json .....	769
Endpoint-Post-/ World/ Infinity/ Get-Color-Palettes→Json .....	740	Endpoint-Post-/ World/ Infinity/ Send-Mail-Message→Json .....	770
Endpoint-Post-/ World/ Infinity/ Get-Inventory→Json .....	742	Endpoint-Post-/ World/ Infinity/ Send-Out-Of-Band-Message→Json .....	771
Endpoint-Post-/ World/ Infinity/ Get-Inventory-By-Type→Json .....	741	Endpoint-Post-/ World/ Infinity/ Server-Time→Json .....	772
Endpoint-Post-/ World/ Infinity/ Get-Mail-In-Box→Json .....	743	Endpoint-Post-/ World/ Infinity/ Set-Avatar-Color→Json .....	773
Endpoint-Post-/ World/ Infinity/ Get-Online-Users→Json .....	744	Endpoint-Post-/ World/ Infinity/ Set-Furniture→Json .....	774
Endpoint-Post-/ World/ Infinity/ Get-Passport→Json .....	745	Endpoint-Post-/ World/ Infinity/ Set-Room-Var→Json .....	775
Endpoint-Post-/ World/ Infinity/ Get-Room-List→Json .....	746	Endpoint-Post-/ World/ Infinity/ Set-User-Var→Json .....	776
Endpoint-Post-/ World/ Infinity/ Get-Room-Vars→Json .....	747	Endpoint-Post-/ World/ Infinity/ Shoot→Json .....	777
Endpoint-Post-/ World/ Infinity/ Get-Server-Time→Json .....	748		

Endpoint-Post-/ World/ Infinity/ Spawn-Zone→Json.....	778
Endpoint-Post-/ World/ Infinity/ Speak→Json.....	779
Endpoint-Post-/ World/ Infinity/ Stamp-Passport→Json.....	780
Endpoint-Post-/ World/ Infinity/ Start-Event→Json.....	781
Endpoint-Post-/ World/ Infinity/ Toot-List→Json.....	782
Endpoint-Post-/ World/ Infinity/ Use-Equipment→Json.....	783
Endpoint-Post-/ World/ Infinity/ Wardrobe→Json.....	784
Endpoint-Post-/ World/ Infinity/ Wtl→Json.....	786
Endpoint-Post-/ World/ Infinity/ Wtl-4→Json.....	785
Endpoint-Put-/ Toots/ Toot-Name→Json.....	788
Endpoint-Put-/ Users/ Me→Json.....	789
Endpoint-Template.....	790
Endpoint-Template-Arity.....	791
Endpoint-Template-Match.....	792
Endpoint-Template-String.....	793
Endpoint-Vars->Openapi.....	794
Endpoints-Equal.....	795
Endpoints-Page-Footer.....	796
Endpoints-Page-Header.....	797
Endpoints-Prefixed.....	798
Enqueue.....	1766
Ensure-Integer.....	799
Ensure-Inventory-Item.....	800
Ensure-List-Of-People.....	801
Ensure-Message-Is-Characters.....	802
Ensure-Number.....	803
Ensure-Record.....	804
Ensure-Site-Name.....	805
Ensure-Toot.....	806
Ensure-User-For-Plist.....	807
Ensure-Wear-Slot.....	808
Ensure-Weather-Kernel.....	809
Entry.....	810
Enumerate-Endpoints.....	811
Erase-All-Memcached-For.....	812
Error!.....	75
Error-Log-File.....	813
Escaped.....	76
Evacuate.....	155
Every-Toot-Name.....	814
Exponent-Digit.....	24
Extension-For-Content-Type.....	815
Extract-Certificate-Base64.....	816
Extract-Plist-Path.....	817
Extract-Public-Key-From-Cert.....	818

**F**

Fetch-Ice-Credentials/ Xirsys.....	819
Fetch-Json.....	820
Fill-Blank-Contour.....	821
Filter.....	156
Find-Acceptor.....	822
Find-Appropriate-Backtrace.....	77
Find-Best-Endpoint.....	823
Find-Client-For-Socket.....	824
Find-Exact-Endpoint.....	825
Find-Infinity-Websocket-Resource.....	826
Find-Kinda-Endpoint.....	827
Find-Log-Dir.....	828
Find-Person-By-Url.....	829
Find-Player-Or-Die.....	830
Find-Random-Point-If.....	831
Find-Record.....	832
Find-Records.....	833
Find-Records-By-Sql.....	834
Find-Reference.....	835
Find-Results-In-Docstring.....	836
Find-Robot.....	837
Find-Thread.....	838
Find-Toot-By-Name.....	839
Find-Toot-Passport.....	840
Find-User-For-Credentials.....	841
Find-User-For-Headers.....	842
Find-User-For-Json.....	843
Find-Var-In-Docstring.....	844
Finger.....	157
First-Line.....	845
First-Paragraph.....	846
First-Weekday-Of-Month.....	25
Flatten-Plist-Tree.....	847
Flush.....	158
Force-Close-Hunchensocket.....	849
Format-Language.....	1767
Format-Symbol-Name-Carefully.....	78
Fountain-Duplicate-P.....	850
Fountain-Reject-As-Already-Done.....	851
From-Avatars.....	852

**G**

Game.....	159
Game-Action-Bowling-Reset-Pins.....	853
Game-Action-Bowling-Strike-Pins.....	854
Game-Action-Card-Game-Arrange.....	855
Game-Action-Card-Game-Deal.....	856
Game-Action-Card-Game-Draw.....	857
Game-Action-Card-Game-Move.....	858
Game-Action-Card-Game-Play.....	859
Game-Action-Card-Game-Shuffle.....	860
Game-Action-Card-Game-Take.....	861
Game-Action-Get-Bowling-Scorecard.....	862
Game-Action-Join-Bowling-Game.....	863
Game-Action-Join-Card-Game.....	864
Game-Action-Part-Bowling-Game.....	866

Game-Action-Part-Card-Game	867	Gracefully-Report-Error.Html	904
Game-Action-Pause-Sports-Ball-Timer	868	Gracefully-Report-Error.Json	905
Game-Action-Sports-Ball-Goal	869	Gracefully-Report-Http-Client-Error	906
Game-Action-Start-Bowling	870	Grant	173
Game-Action-Start-Sports-Ball-Game	871	Grant-Item	907
Game-Action-Start-Sports-Ball-Timer	872	Gravatar-Hash	908
Game-Action-Tag-You-Re-It	873	Gravatar-Image-Url	909
Game-Point-Altitude	875	Greeting/ Daemon/ Error-Output	910
Game-Point-Altitude, SetF	875	Greeting/ Daemon/ Log-Output	911
Game-Point-Latitude	876	Greeting/ Daemon/ Standard-Output	912
Game-Point-Latitude, SetF	876	Greeting/ Daemon/ Trace-Output	913
Game-Point-Longitude	877	Group-Plists	914
Game-Point-Longitude, SetF	877		
Game-Point-P	878	<b>H</b>	
Game-Point-World	879	Habitat-Elevation-Roughness	915
Game-Point-World, SetF	879	Habitat<-Pixel	916
Game-Point-X	880	Handle-Normal-Request	917
Game-Point-X, SetF	880	Handle-Options-Request	918
Game-Point-Y	881	Hangup	1768
Game-Point-Y, SetF	881	Headcount	174
Game-Point-Z	882	Header-Time	920
Game-Point-Z, SetF	882	Holiday-On	26
Gather-All-Symbols	883	Host-Name-Char-P	922
Gather-Source-Info	79	Host-Name-Like-P	923
Gc	160	How-Slow-Is-Slow	924
Generate-Blank-Contour	884	Http-Error-Got-Uri	81
Generate-Buddy-List-Signature	885	Http-Error-Headers	82
Generate-Skydome-Cloud-Layer	886	Http-Error-Status	83
Generate-Terrain-Blank-Edge-Horz	887	Http-Error-Status-Text	84
Generate-Terrain-Blank-Edge-Vert	888	Http-Error-Wanted-Uri	85
Generate-Terrain-Contour	889	Http-Is-Success-P	927
Generate-Terrain-Features	890	Http-Status-Code	930
Get-9-Terrain-Tiles	891	Http-Successful-Request	86
Get-Google-Account-Keys	892		
Get-Last-Insert-Id	893	<b>I</b>	
Get-Mariadb-Lock	894	Ice-Credentials	931
Get-Rollbar-Person	895	Ice-Url-To-Urals	932
Get-Unix-Time	896	Id-Column-For	933
Getconfig	161	Ignore-Duplicates	934
Getevents	162	Ignore-Not-Found	935
Getmotd	163	Infinity-Add-Furniture	936
Getschedule	164	Infinity-Add-Journal-Entry	937
Getschedulefor	165	Infinity-Add-To-List	938
Getuvar	166	Infinity-Click	939
Getuvars	167	Infinity-Consider-Child-Approval	941
Getvar	168	Infinity-Create-User-House	942
Getvars	169	Infinity-Delete-Mail-Message	943
Gift-Item	897	Infinity-Doff	944
Git-Pull	170	Infinity-Dofff	945
Give	171	Infinity-Don	946
Givehead	172	Infinity-Echo	948
Global-Heightmap-Corner	898	Infinity-End-Event	949
Global-Heightmap-Corner, SetF	898	Infinity-Enumerate-Wear-Slots	952
Gossip-Initiation-Answer	901	Infinity-Finger	953
Gossip-Initiation-Answer, SetF	901	Infinity-Game-Action	954
Gossip-Initiation-P	902	Infinity-Get-Apple	956
Gossip-Initiation-Uuid	903		
Gossip-Initiation-Uuid, SetF	903		

Infinity-Get-Avatars .....	959	Invalidate-Cache .....	1036
Infinity-Get-Color-Palettes .....	960	Inventory-Item-Equipped .....	1038
Infinity-Get-Inventory .....	961	Inventory-Item-Equipped, SetF .....	1038
Infinity-Get-Inventory-By-Type .....	962	Inventory-Item-Equipped-P .....	1039
Infinity-Get-Mail-In-Box .....	964	Inventory-Item-Item .....	1040
Infinity-Get-Online-Users .....	966	Inventory-Item-Item, SetF .....	1040
Infinity-Get-Passport .....	967	Inventory-Item-P .....	1041
Infinity-Get-Room-List .....	968	Inventory-Item-Person .....	1042
Infinity-Get-Room-Vars .....	969	Inventory-Item-Person, SetF .....	1042
Infinity-Get-Server-Time .....	973	Inventory-Item-Toot .....	1043
Infinity-Get-Session-Apple .....	974	Inventory-Item-Toot, SetF .....	1043
Infinity-Get-Store-Item-Info .....	975	Item-Alt-Color .....	1045
Infinity-Get-User-Lists .....	976	Item-Alt-Color, SetF .....	1045
Infinity-Get-Wallet .....	977	Item-Altitude .....	1046
Infinity-Get-Zone-List .....	978	Item-Altitude, SetF .....	1046
Infinity-Give .....	979	Item-Avatar-Scale-X .....	1047
Infinity-Go .....	980	Item-Avatar-Scale-X, SetF .....	1047
Infinity-Init-User-Room .....	981	Item-Avatar-Scale-Y .....	1048
Infinity-Join .....	982	Item-Avatar-Scale-Y, SetF .....	1048
Infinity-Login .....	983	Item-Avatar-Scale-Z .....	1049
Infinity-Logout .....	985	Item-Avatar-Scale-Z, SetF .....	1049
Infinity-Mail-Customer-Service .....	986	Item-Base-Color .....	1050
Infinity-Peek-At-Inventory .....	987	Item-Base-Color, SetF .....	1050
Infinity-Ping .....	988	Item-Energy .....	1051
Infinity-Play-With .....	989	Item-Energy, SetF .....	1051
Infinity-Pre-Login .....	990	Item-Facing .....	1052
Infinity-Prompt-Reply .....	991	Item-Facing, SetF .....	1052
Infinity-Quiesce .....	994	Item-Gain-Energy .....	1053
Infinity-Read-Map .....	995	Item-In-Inventory-P .....	1054
Infinity-Remove-From-List .....	996	Item-Info .....	1055
Infinity-Report-Bug .....	997	Item-Latitude .....	1056
Infinity-Report-User .....	1002	Item-Latitude, SetF .....	1056
Infinity-Request-Buddy .....	1003	Item-Longitude .....	1057
Infinity-Send-Mail-Message .....	1004	Item-Longitude, SetF .....	1057
Infinity-Send-Out-Of-Band-Message .....	1006	Item-Lose-Energy .....	1058
Infinity-Server-Time .....	1007	Item-Owned-By-P .....	1059
Infinity-Set-Avatar-Color .....	1008	Item-P .....	1060
Infinity-Set-Furniture .....	1009	Item-Template .....	1061
Infinity-Set-Room-Var .....	1011	Item-Template, SetF .....	1061
Infinity-Set-User-Var .....	1012	Item-Template-Avatar .....	1062
Infinity-Shoot .....	1014	Item-Template-Avatar, SetF .....	1062
Infinity-Spawn-Zone .....	1015	Item-Template-Avatar-Scale-X .....	1063
Infinity-Speak .....	1016	Item-Template-Avatar-Scale-X, SetF .....	1063
Infinity-Stamp-Passport .....	1019	Item-Template-Avatar-Scale-Y .....	1064
Infinity-Start-Event .....	1020	Item-Template-Avatar-Scale-Y, SetF .....	1064
Infinity-Stats .....	175, 1023	Item-Template-Avatar-Scale-Z .....	1065
Infinity-Toot-List .....	1024	Item-Template-Avatar-Scale-Z, SetF .....	1065
Infinity-Use-Equipment .....	1025	Item-Template-Default-Alt-Color .....	1066
Infinity-Wardrobe .....	1026	Item-Template-Default-Alt-Color, SetF .....	1066
Infinity-Wtl .....	1028	Item-Template-Default-Base-Color .....	1067
Infinity-Wtl-4 .....	1030	Item-Template-Default-Base-Color, SetF .....	1067
Info! .....	87	Item-Template-Default-Base-Color, SetF .....	1068
Init-Async .....	1031	Item-Template-Description .....	1068
Init-Characters .....	1032	Item-Template-Description, SetF .....	1068
Integer-To-Byte-Vector .....	1033	Item-Template-Energy-Kind .....	1069
Integer-To-Color24 .....	1034	Item-Template-Energy-Kind, SetF .....	1069
Interpret-Facing .....	1035	Item-Template-Energy-Max .....	1070
Inv .....	176	Item-Template-Energy-Max, SetF .....	1070
		Item-Template-Energy-Max, SetF .....	1070
		Item-Template-Id .....	1071

Item-Template-Id, SetF	1071	Locale-Music-P	1107
Item-Template-Info	1072	Locale-Music-Radius	1108
Item-Template-Name	1073	Locale-Music-Radius, SetF	1108
Item-Template-Name, SetF	1073	Locale-Music-X	1109
Item-Template-On-Zero	1074	Locale-Music-X, SetF	1109
Item-Template-On-Zero, SetF	1074	Locale-Music-Y	1110
Item-Template-P	1075	Locale-Music-Y, SetF	1110
Item-Template-Trade	1076	Locale-Music-Z	1111
Item-Template-Trade, SetF	1076	Locale-Music-Z, SetF	1111
Item-Template-Wear-Slot	1077	Login-Child	1113
Item-Template-Wear-Slot, SetF	1077	Login-Credential	1114
Item-Template-Weight	1078	Login-Credential, SetF	1114
Item-Template-Weight, SetF	1078	Login-Fail	1115
Item-Uuid	1079	Login-Failed-Message	1116
Item-Uuid, SetF	1079	Login-Last-Seen	1117
Item-World	1080	Login-Last-Seen, SetF	1117
Item-World, SetF	1080	Login-Ok-Message	1118
Item-X	1081	Login-Origin	1119
Item-X, SetF	1081	Login-Origin, SetF	1119
Item-Y	1082	Login-P	1120
Item-Y, SetF	1082	Login-Person	1121
Item-Z	1083	Login-Person, SetF	1121
Item-Z, SetF	1083	Login-Renewed	1122
Items-At	1084	Login-Renewed, SetF	1122
		Login-Start	1123
		Login-Start, SetF	1123
		Login-Uuid	1124
		Login-Uuid, SetF	1124
		Longitude	1125
		Look-For-Ssl-Certs	1126
		Lot-Owner-Toot	1128
		Lot-Owner-Toot, SetF	1128
		Lot-Ownership	1129
		Lot-Ownership, SetF	1129
		Lot-P	1130
		Lot-World	1131
		Lot-World, SetF	1131
		Lot-X1	1132
		Lot-X1, SetF	1132
		Lot-X2	1133
		Lot-X2, SetF	1133
		Lot-Y1	1134
		Lot-Y1, SetF	1134
		Lot-Y2	1135
		Lot-Y2, SetF	1135
		Lot-Z1	1136
		Lot-Z1, SetF	1136
		Lot-Z2	1137
		Lot-Z2, SetF	1137
<b>J</b>			
Journal	1086		
Json-To-Html	1087		
<b>K</b>			
Kick	177, 1088		
Kick-Child-Time-Up	1089		
King	179		
<b>L</b>			
Lambda-List-As-Variables	1091		
Last-Active	1092		
Last-Active, SetF	1092		
Latitude	1093		
Leave	1769		
Legal-Age	1094		
Level-Is-Valid-P	88		
Liftban	180		
Limit-String-Length	1096		
Lisp-To-Db-Name	1097		
List-Banhammers	1098		
List-Of-String=	1099		
Listen-For-Websockets	1100		
Listener-Name	1101		
Load-Config	1102		
Load-Record	1103		
Loadlists	181		
Local-Room-Vars	1104		
Locale-Music-Music	1106		
Locale-Music-Music, SetF	1106		

## M

Make-Avatar.....	1138
Make-Avatar-Slot.....	1139
Make-Character-Music.....	1140
Make-Child-Request.....	1141
Make-Color24.....	1142
Make-Contact.....	1143
Make-Credential.....	1144
Make-Endpoint-Function-Name.....	1145
Make-Game-Point.....	1146
Make-Gossip-Initiation.....	1147
Make-Inventory-Item.....	1148
Make-Item.....	1149
Make-Item-Template.....	1150
Make-Level-Notifier.....	89
Make-Locale-Music.....	1151
Make-Login.....	1152
Make-Lot.....	1153
Make-Metronome-Task.....	1154
Make-Mist.....	1155
Make-Music.....	1156
Make-New-Toot-State.....	1157
Make-Parent-Child.....	1158
Make-Pattern.....	1159
Make-Person.....	1160
Make-Person-Link.....	1161
Make-Place.....	1162
Make-Quaestor-Event.....	1163
Make-Record.....	1164
Make-Sms.....	1165
Make-Store-Item.....	1166
Make-Tcp-Client.....	1167
Make-Terrain-Height.....	1168
Make-Thread-Name.....	116
Make-Toot.....	1169
Make-Toot-Quiesced.....	1170
Make-Wear-Slot.....	1171
Make-Wind-Vector.....	1172
Make-Wind-Vector-Field.....	1173
Make-World.....	1174
Make-Wtl-Course.....	1175
Maybe-Parent-Approval.....	1177
Mem.....	182
Memcached-Get-Key.....	1179
Message.....	1770
Metronome.....	183
Metronome-Idle-Tasks.....	1180
Metronome-Register.....	1181
Metronome-Remove.....	1182
Metronome-Task-Frequency.....	1184
Metronome-Task-Frequency, SetF.....	1184
Metronome-Task-Function.....	1185
Metronome-Task-Function, SetF.....	1185
Metronome-Task-Name.....	1186
Metronome-Task-Name, SetF.....	1186
Metronome-Task-One-Shot-Time.....	1187
Metronome-Task-One-Shot-Time, SetF.....	1187
Metronome-Task-P.....	1188
Metronome-Task-Thread.....	1189
Metronome-Task-Thread, SetF.....	1189
Mist-Altitude-1.....	1191
Mist-Altitude-1, SetF.....	1191
Mist-Altitude-2.....	1192
Mist-Altitude-2, SetF.....	1192
Mist-Definedp.....	1193
Mist-Definedp, SetF.....	1193
Mist-Latitude-1.....	1194
Mist-Latitude-1, SetF.....	1194
Mist-Latitude-2.....	1195
Mist-Latitude-2, SetF.....	1195
Mist-Longitude-1.....	1196
Mist-Longitude-1, SetF.....	1196
Mist-Longitude-2.....	1197
Mist-Longitude-2, SetF.....	1197
Mist-P.....	1198
Mist-World.....	1199
Mist-World, SetF.....	1199
Month*.....	28
Moon-Position.....	1201
Motd.....	184
Music-Artist.....	1203
Music-Artist, SetF.....	1203
Music-Genre.....	1204
Music-Genre, SetF.....	1204
Music-Id.....	1205
Music-Id, SetF.....	1205
Music-License.....	1206
Music-License, SetF.....	1206
Music-Moniker.....	1207
Music-Moniker, SetF.....	1207
Music-P.....	1208
Music-Title.....	1209
Music-Title, SetF.....	1209
Mute.....	185
<b>N</b>	
Name-For-Content-Type.....	1210
Name-Idle-Threads-Sequentially.....	117, 1211
Named-Thread-Pool-Runner.....	118
Nearp.....	1212
Normalize-Url.....	1214
Not-Found-If-Null.....	1216
Not-Found-Thing.....	1217
Not-Found-Thing, SetF.....	1217
Notify.....	90
Nuke.....	186
Null-If-Empty.....	1219
<b>O</b>	
On-Exception.....	1220
Open-Log-File.....	1221
Output-For-Level.....	91



## P

Package-Name-Can-Be-Unquoted-P .....	92
Pad-To-Multiple-Of-8 .....	1222
Parent-Child-Child .....	1224
Parent-Child-Child, SetF .....	1224
Parent-Child-P .....	1225
Parent-Child-Parent .....	1226
Parent-Child-Parent, SetF .....	1226
Parent-Deny-Permission .....	1227
Parent-Grant-Permission .....	1228
Parentapproves .....	187
Parse-Backtrace .....	1229
Parse-Color24 .....	1230
Parse-Operator-Command .....	1231
Parse-Uri-As-Template .....	1232
Parse-Wtl-For-Robot .....	1233
Path->Openapi .....	1234
Pattern-Id .....	1236
Pattern-Id, SetF .....	1236
Pattern-Name .....	1237
Pattern-Name, SetF .....	1237
Pattern-P .....	1238
Pause .....	1771
Peer-Address .....	1239
Pending-Child-Approval-Request .....	1240
Pending-Child-Requests-By-Toot .....	1241
Person-Age .....	1243
Person-Age* .....	1244
Person-Age, SetF .....	1243
Person-Date-Of-Birth .....	1245
Person-Date-Of-Birth, SetF .....	1245
Person-Display-Name .....	1246
Person-Display-Name, SetF .....	1246
Person-First-Email .....	1247
Person-Gender .....	1248
Person-Gender, SetF .....	1248
Person-Given-Name .....	1249
Person-Given-Name, SetF .....	1249
Person-Info .....	1250
Person-Is-Patron-P .....	1251
Person-Lang .....	1252
Person-Lang, SetF .....	1252
Person-Link-Label .....	1254
Person-Link-Label, SetF .....	1254
Person-Link-P .....	1255
Person-Link-Person .....	1256
Person-Link-Person, SetF .....	1256
Person-Link-Provenance .....	1257
Person-Link-Provenance, SetF .....	1257
Person-Link-Rel .....	1258
Person-Link-Rel, SetF .....	1258
Person-Link-Url .....	1259
Person-Link-Url, SetF .....	1259
Person-Link-Uuid .....	1260
Person-Link-Uuid, SetF .....	1260
Person-Links-To-Email .....	1261
Person-P .....	1262
Person-Sensitivep .....	1263
Person-Sensitivep, SetF .....	1263
Person-Surname .....	1264
Person-Surname, SetF .....	1264
Person-Uuid .....	1265
Person-Uuid, SetF .....	1265
Ping .....	188
Place .....	189
Place-Altitude .....	1268
Place-Altitude, SetF .....	1268
Place-Appearance .....	1269
Place-Appearance, SetF .....	1269
Place-Attributes .....	1270
Place-Attributes, SetF .....	1270
Place-Furniture .....	1271
Place-Kind .....	1272
Place-Kind, SetF .....	1272
Place-Latitude .....	1273
Place-Latitude, SetF .....	1273
Place-Longitude .....	1274
Place-Longitude, SetF .....	1274
Place-P .....	1275
Place-Shape .....	1276
Place-Shape, SetF .....	1276
Place-String .....	1277
Place-String-Circle .....	1278
Place-Uuid .....	1279
Place-Uuid, SetF .....	1279
Place-World .....	1280
Place-World, SetF .....	1280
Places-At-Position .....	1281
Play .....	1772
Play-Digits .....	1773
Play-With-Toot .....	1282
Player-Adultp .....	1283
Player-Alert .....	1284
Player-Childp .....	1285
Player-Toots .....	1286
Plist-To-English .....	1287
Plist-With-Index .....	1288
Point-Underwater-P .....	1289
Post-Sign-In .....	1290
Post/ Read-Version-Page .....	1291
Potential-Toot-Name-Character-P .....	1292
Potential-Toot-Name-P .....	1293
Power-On-Self-Test .....	1294
Powerset .....	1295
Pre-Login-Commands .....	1296
Pre-Login-Commands, SetF .....	1296
Precipitation .....	1297
Pretty-Function-Name .....	93
Pretty-Print-Html-Error .....	1298
Pretty-Symbol-Name .....	94
Print-Help .....	1299
Private-Admin-Message .....	1300
Pull-Records .....	1303
Purgephysics .....	193
Push-Script .....	194

## Q

Quaestor-Cancel-Event .....	1306
Quaestor-Complete-Event .....	1307
Quaestor-End-Fountain .....	1308
Quaestor-Event-Completedp .....	1310
Quaestor-Event-Completedp, SetF .....	1310
Quaestor-Event-Ended-At .....	1311
Quaestor-Event-Ended-At, SetF .....	1311
Quaestor-Event-Fairy-Dust .....	1312
Quaestor-Event-Fairy-Dust, SetF .....	1312
Quaestor-Event-Item .....	1313
Quaestor-Event-Item, SetF .....	1313
Quaestor-Event-Medal .....	1314
Quaestor-Event-Medal, SetF .....	1314
Quaestor-Event-P .....	1315
Quaestor-Event-Peanuts .....	1316
Quaestor-Event-Peanuts, SetF .....	1316
Quaestor-Event-Score .....	1317
Quaestor-Event-Score, SetF .....	1317
Quaestor-Event-Source .....	1318
Quaestor-Event-Source, SetF .....	1318
Quaestor-Event-Started-At .....	1319
Quaestor-Event-Started-At, SetF .....	1319
Quaestor-Event-Started-By .....	1320
Quaestor-Event-Started-By, SetF .....	1320
Quaestor-Event-Uuid .....	1321
Quaestor-Event-Uuid, SetF .....	1321
Quaestor-New-Toot .....	1322
Quaestor-Start-Event .....	1323
Quaestor-Start-General .....	1324
Query-Params .....	1325
Query-String->Plist .....	1326
Query-To-Memcache-Key .....	1327
Quick-Reload .....	195
Quiesce-Connected-Toots .....	1328
Quoted .....	95

## R

Random-Key .....	1330
Random-Key, SetF .....	1330
Random-Start-Wtl-For-Toot .....	1331
Raw-Post-String .....	1332
Rc .....	196
Read-Related-Journal .....	1333
Read-Staff-Journal .....	1334
Reap-Uninteresting-Child-Requests .....	1335
Reasonable-Name-Char-P .....	1336
Reasonable-Name-P .....	1337
Reboot .....	197
Rebuild-Myself .....	1338
Record .....	1774
Redact-Directory .....	96
Redirect .....	1775
Redirect-To .....	1339
Redirect-To/ Html-Body .....	1340
Register-Dns-Name .....	54
Register-Metronome-Tasks .....	1341

Register-Signal-Handlers .....	1342
Reject .....	1776
Relative-Facing .....	1343
Reload-Production .....	1344
Reloadconfig .....	198
Remap-Endpoints .....	1345
Remove-Furniture .....	1346
Remove-Repeats-For-Toot-Name .....	1347
Rename-Toot .....	1348
Render-Json .....	1349
Replace-TeXinfo-Tables .....	1350
Report-Server-Info .....	97
Report-Slow-Query .....	1351
Report-Telemetry .....	98
Request-Accept-Types .....	1352
Request-Telemetry .....	99
Respond-To-Error .....	1353
Restore-Robot-Wtl .....	1354
Retire .....	199
Return-New-Apple .....	1355
Rgb-Bytes->Rgb .....	1356
Robot-Broadcast .....	1358
Robot-Course .....	1360
Robot-Course, SetF .....	1360
Robot-Course-Wtl .....	1361
Robot-Go-To .....	1366
Robot-Handle .....	1367
Robot-Has-Heard .....	1369
Robot-Has-Heard, SetF .....	1369
Robot-Heard .....	1370
Robot-Listen .....	1373
Robot-Match .....	1374
Robot-Mode .....	1376
Robot-Mode, SetF .....	1376
Robot-Position .....	1380
Robot-Say .....	1383
Robot-Set-Mode .....	1384
Robot-Unicast .....	1391
Robotp .....	1393
Rollbar-Notify-Deployment .....	100
Romance-Ii-Copyright-Latest .....	1394
Romance-Ii-Program-Name .....	1395
Romance-Ii-Program-Name/ Version .....	1396
Romance-Ii-Program-Version .....	1397
Run .....	200
Run-Async .....	1398
Run-Metronome-Tasks .....	1399

## S

Safe-Client-As-String .....	119	SetF Credential-Provider .....	563
Sanitize-File-Name .....	101	SetF Credential-Refresh-Token .....	564
Save-Record .....	1400	SetF Credential-Uid .....	565
Saveroomvars .....	201	SetF Credential-Uuid .....	566
Say .....	1777	SetF Game-Point-Altitude .....	875
Scotty .....	202	SetF Game-Point-Latitude .....	876
Script .....	203	SetF Game-Point-Longitude .....	877
Send-Parent-Child-Login-Email .....	1401	SetF Game-Point-World .....	879
Send-Parent-Child-Login-Request .....	1402	SetF Game-Point-X .....	880
Send-Reply-As-Bytes .....	1403	SetF Game-Point-Y .....	881
Send-Rollbar-Notification .....	102	SetF Game-Point-Z .....	882
Send-Sms-Message .....	1404	SetF Global-Heightmap-Corner .....	898
Server-List .....	204, 1405	SetF Gossip-Initiation-Answer .....	901
Set-Http-Default-Headers .....	1406	SetF Gossip-Initiation-Uuid .....	903
Set-Up-For-Daemon/ Error-Output .....	1407	SetF Inventory-Item-Equipped .....	1038
Set-Up-For-Daemon/ Log-Output .....	1408	SetF Inventory-Item-Item .....	1040
Set-Up-For-Daemon/ Standard-Output .....	1409	SetF Inventory-Item-Person .....	1042
Set-Up-For-Daemon/ Start-Logging .....	1410	SetF Inventory-Item-Toot .....	1043
Set-Up-For-Daemon/ Trace-Output .....	1411	SetF Item-Alt-Color .....	1045
Set-User-Var .....	1412	SetF Item-Altitude .....	1046
Set-User-Var-D .....	1413	SetF Item-Avatar-Scale-X .....	1047
Set-User-Var-Wtl .....	1414	SetF Item-Avatar-Scale-Y .....	1048
Setavatarcolors .....	205	SetF Item-Avatar-Scale-Z .....	1049
Setbadge .....	206	SetF Item-Base-Color .....	1050
Setconfig .....	207	SetF Item-Energy .....	1051
SetF Avatar-Avatar-Scale-X .....	390	SetF Item-Facing .....	1052
SetF Avatar-Avatar-Scale-Y .....	391	SetF Item-Latitude .....	1056
SetF Avatar-Avatar-Scale-Z .....	392	SetF Item-Longitude .....	1057
SetF Avatar-Id .....	394	SetF Item-Template .....	1061
SetF Avatar-Moniker .....	395	SetF Item-Template-Avatar .....	1062
SetF Avatar-Slot-Avatar .....	398	SetF Item-Template-Avatar-Scale-X .....	1063
SetF Avatar-Slot-Id .....	399	SetF Item-Template-Avatar-Scale-Y .....	1064
SetF Avatar-Slot-Slot .....	401	SetF Item-Template-Avatar-Scale-Z .....	1065
SetF Avatar-Slot-Valence .....	402	SetF Item-Template-Default-Alt-Color .....	1066
SetF Bad-Request-Thing .....	407	SetF Item-Template-Default-Base-Color .....	1067
SetF Character-Music-Music .....	438	SetF Item-Template-Description .....	1068
SetF Character-Music-Toot .....	440	SetF Item-Template-Energy-Kind .....	1069
SetF Child-Request-Allowed-At .....	455	SetF Item-Template-Energy-Max .....	1070
SetF Child-Request-Allowed-For .....	456	SetF Item-Template-Id .....	1071
SetF Child-Request-Denied-At .....	458	SetF Item-Template-Name .....	1073
SetF Child-Request-Placed-At .....	460	SetF Item-Template-On-Zero .....	1074
SetF Child-Request-Response .....	461	SetF Item-Template-Trade .....	1076
SetF Child-Request-Toot .....	462	SetF Item-Template-Wear-Slot .....	1077
SetF Child-Request-Uuid .....	463	SetF Item-Template-Weight .....	1078
SetF Color24-Blue .....	478	SetF Item-Uuid .....	1079
SetF Color24-Green .....	479	SetF Item-World .....	1080
SetF Color24-Red .....	484	SetF Item-X .....	1081
SetF Contact-Added .....	512	SetF Item-Y .....	1082
SetF Contact-Contact .....	513	SetF Item-Z .....	1083
SetF Contact-Last-Used .....	514	SetF Last-Active .....	1092
SetF Contact-Owner .....	515	SetF Locale-Music-Music .....	1106
SetF Contact-Starredp .....	517	SetF Locale-Music-Radius .....	1108
SetF Contact-Uuid .....	518	SetF Locale-Music-X .....	1109
SetF Credential-Auth-Token .....	558	SetF Locale-Music-Y .....	1110
SetF Credential-Id-Token .....	559	SetF Locale-Music-Z .....	1111
SetF Credential-Json-Info .....	560	SetF Login-Credential .....	1114
SetF Credential-Person .....	562	SetF Login-Last-Seen .....	1117
		SetF Login-Origin .....	1119

SetF Login-Person .....	1121	SetF Place-Shape .....	1276
SetF Login-Renewed .....	1122	SetF Place-Uuid .....	1279
SetF Login-Start .....	1123	SetF Place-World .....	1280
SetF Login-Uuid .....	1124	SetF Pre-Login-Commands .....	1296
SetF Lot-Owner-Toot .....	1128	SetF Quaestor-Event-Completedp .....	1310
SetF Lot-Ownership .....	1129	SetF Quaestor-Event-Ended-At .....	1311
SetF Lot-World .....	1131	SetF Quaestor-Event-Fairy-Dust .....	1312
SetF Lot-X1 .....	1132	SetF Quaestor-Event-Item .....	1313
SetF Lot-X2 .....	1133	SetF Quaestor-Event-Medal .....	1314
SetF Lot-Y1 .....	1134	SetF Quaestor-Event-Peanuts .....	1316
SetF Lot-Y2 .....	1135	SetF Quaestor-Event-Score .....	1317
SetF Lot-Z1 .....	1136	SetF Quaestor-Event-Source .....	1318
SetF Lot-Z2 .....	1137	SetF Quaestor-Event-Started-At .....	1319
SetF Metronome-Task-Frequency .....	1184	SetF Quaestor-Event-Started-By .....	1320
SetF Metronome-Task-Function .....	1185	SetF Quaestor-Event-Uuid .....	1321
SetF Metronome-Task-Name .....	1186	SetF Random-Key .....	1330
SetF Metronome-Task-One-Shot-Time .....	1187	SetF Robot-Course .....	1360
SetF Metronome-Task-Thread .....	1189	SetF Robot-Has-Heard .....	1369
SetF Mist-Altitude-1 .....	1191	SetF Robot-Mode .....	1376
SetF Mist-Altitude-2 .....	1192	SetF Sms-Destination .....	1426
SetF Mist-Definedp .....	1193	SetF Sms-Message .....	1427
SetF Mist-Latitude-1 .....	1194	SetF Sms-Mmsp .....	1429
SetF Mist-Latitude-2 .....	1195	SetF Sms-Sender .....	1431
SetF Mist-Longitude-1 .....	1196	SetF Sms-Uuid .....	1432
SetF Mist-Longitude-2 .....	1197	SetF Store-Item-Currency .....	1458
SetF Mist-World .....	1199	SetF Store-Item-Price .....	1460
SetF Music-Artist .....	1203	SetF Store-Item-Qty .....	1461
SetF Music-Genre .....	1204	SetF Store-Item-Template .....	1462
SetF Music-Id .....	1205	SetF Store-Item-Uuid .....	1463
SetF Music-License .....	1206	SetF Taskmaster-Thread-Pool .....	121
SetF Music-Moniker .....	1207	SetF Taskmaster-Thread-Pool-Channel .....	122
SetF Music-Title .....	1209	SetF Tcp-Client-Buffer .....	1476
SetF Not-Found-Thing .....	1217	SetF Tcp-Client-Expected-Length .....	1477
SetF Parent-Child-Child .....	1224	SetF Tcp-Client-Peer .....	1479
SetF Parent-Child-Parent .....	1226	SetF Tcp-Client-Socket .....	1480
SetF Pattern-Id .....	1236	SetF Terrain-Height-Latitude .....	1494
SetF Pattern-Name .....	1237	SetF Terrain-Height-Longitude .....	1495
SetF Person-Age .....	1243	SetF Terrain-Height-Terrain .....	1497
SetF Person-Date-Of-Birth .....	1245	SetF Terrain-Height-World .....	1498
SetF Person-Display-Name .....	1246	SetF Toot .....	1515
SetF Person-Gender .....	1248	SetF Toot-Avatar .....	1516
SetF Person-Given-Name .....	1249	SetF Toot-Avatar-Scale-X .....	1517
SetF Person-Lang .....	1252	SetF Toot-Avatar-Scale-Y .....	1518
SetF Person-Link-Label .....	1254	SetF Toot-Avatar-Scale-Z .....	1519
SetF Person-Link-Person .....	1256	SetF Toot-Base-Color .....	1520
SetF Person-Link-Provenance .....	1257	SetF Toot-Child-Code .....	1527
SetF Person-Link-Rel .....	1258	SetF Toot-Last-Active .....	1539
SetF Person-Link-Url .....	1259	SetF Toot-Name .....	1541
SetF Person-Link-Uuid .....	1260	SetF Toot-Note .....	1542
SetF Person-Sensitivep .....	1263	SetF Toot-Pad-Color .....	1544
SetF Person-Surname .....	1264	SetF Toot-Pattern .....	1549
SetF Person-Uuid .....	1265	SetF Toot-Pattern-Color .....	1550
SetF Place-Altitude .....	1268	SetF Toot-Player .....	1557
SetF Place-Appearance .....	1269	SetF Toot-Position .....	1558
SetF Place-Attributes .....	1270	SetF Toot-Quiesced-Altitude .....	1562
SetF Place-Kind .....	1272	SetF Toot-Quiesced-Attribs .....	1563
SetF Place-Latitude .....	1273	SetF Toot-Quiesced-D3 .....	1564
SetF Place-Longitude .....	1274	SetF Toot-Quiesced-Emotion .....	1566

SetF Toot-Quiesced-Latitude.....	1567	Spawn-Terrain.....	1436
SetF Toot-Quiesced-Longitude.....	1568	Spawnroom.....	213
SetF Toot-Quiesced-Observed.....	1569	Spawnzone.....	214
SetF Toot-Quiesced-Peer-Address.....	1571	Speak.....	215
SetF Toot-Quiesced-Toot.....	1572	Split-Backtrace.....	1437
SetF Toot-Quiesced-World.....	1573	Split-Plist.....	1438
SetF Toot-Quiesced-Wtl.....	1574	Square.....	1440
SetF Toot-Uuid.....	1577	Ssl-Certificate.....	1441
SetF Unimplemented-Feature.....	1590	Ssl-Private-Key.....	1442
SetF User-Account.....	1598	Stamp-Toot-Passport.....	1443
SetF Wear-Slot-Alternate.....	1622	Standard-Log-File.....	1444
SetF Wear-Slot-Avatar-Point.....	1623	Start.....	1445
SetF Wear-Slot-Id.....	1624	Start-Game-Metronome.....	1446
SetF Wear-Slot-Name.....	1626	Start-Hunchentoot.....	1447
SetF Wear-Slot-Obstruct-Max.....	1627	Start-Production.....	1448
SetF Wear-Slot-Obstruct-Min.....	1628	Start-Purchase-Event.....	1449
SetF Wear-Slot-Obstruct-Point.....	1629	Start-Swank.....	1450
SetF Wear-Slot-Valence.....	1631	Start-Tcp-Listener.....	1451
SetF Which-Toot-Is-Not-Yours.....	1635	Stfu.....	216
SetF Wind-Vector-X-Magnitude.....	1641	Stop.....	1452
SetF Wind-Vector-Y-Magnitude.....	1642	Stop-Game-Metronome.....	1453
SetF World-Moniker.....	1664	Stop-Listening-For-Websockets.....	1454
SetF World-Name.....	1666	Stop-Production.....	1455
SetF Wtl-Course-Altitude.....	1693	Store-Info.....	1456
SetF Wtl-Course-End-Point.....	1694	Store-Item-Currency.....	1458
SetF Wtl-Course-End-Time.....	1695	Store-Item-Currency, SetF.....	1458
SetF Wtl-Course-Facing.....	1696	Store-Item-P.....	1459
SetF Wtl-Course-Latitude.....	1697	Store-Item-Price.....	1460
SetF Wtl-Course-Longitude.....	1698	Store-Item-Price, SetF.....	1460
SetF Wtl-Course-Speed.....	1700	Store-Item-Qty.....	1461
SetF Wtl-Course-Start-Point.....	1701	Store-Item-Qty, SetF.....	1461
SetF Wtl-Course-Start-Time.....	1702	Store-Item-Template.....	1462
SetF Wtl-Course-World.....	1703	Store-Item-Template, SetF.....	1462
Setstafflevel.....	208	Store-Item-Uuid.....	1463
Setuvar.....	209	Store-Item-Uuid, SetF.....	1463
Setvar.....	210	String-All-Alpha-Chars-P.....	1464
Sha1-Hash.....	1415	String-Length-2-P.....	1465
Sha1-Hex.....	1416	Strip-After-Sem.....	1466
Shanghai.....	211	Subheader-Field.....	1467
Shift-Contour-Point.....	1419	Sun-Position.....	1468
Shout.....	212	Swank-Connected-P.....	120, 1470
Sinus.....	1420	Symbol-Is-Exported-P.....	103
Sky-Contents.....	1421	Symbol-Name-Can-Be-Unquoted-P.....	104
Sky-Room-Var.....	1422	Sync.....	1471
Slot-Values.....	1423		
Smoothe-Contour-200×200.....	1424	<b>T</b>	
Sms-Destination.....	1426	Take-Item.....	1472
Sms-Destination, SetF.....	1426	Taskmaster-Thread-Pool.....	121
Sms-Message.....	1427	Taskmaster-Thread-Pool, SetF.....	121
Sms-Message, SetF.....	1427	Taskmaster-Thread-Pool-Channel.....	122
Sms-Message-Index.....	1428	Taskmaster-Thread-Pool-Channel, SetF.....	122
Sms-Mmsp.....	1429	Tcp-Bandwidth-Record.....	1473
Sms-Mmsp, SetF.....	1429	Tcp-Broadcast.....	1474
Sms-P.....	1430	Tcp-Client-Buffer.....	1476
Sms-Sender.....	1431	Tcp-Client-Buffer, SetF.....	1476
Sms-Sender, SetF.....	1431	Tcp-Client-Expected-Length.....	1477
Sms-Uuid.....	1432	Tcp-Client-Expected-Length, SetF.....	1477
Sms-Uuid, SetF.....	1432		

Tcp-Client-P	1478	Toot-Can-Afford-P	1524
Tcp-Client-Peer	1479	Toot-Chat-Background-Color	1525
Tcp-Client-Peer, SetF	1479	Toot-Chat-Foreground-Color	1526
Tcp-Client-Socket	1480	Toot-Child-Code	1527
Tcp-Client-Socket, SetF	1480	Toot-Child-Code, SetF	1527
Tcp-Format-Error	1481	Toot-Childp	1528
Tcp-Handle-Peer-Request	1482	Toot-Clothes+Pattern	1529
Tcp-Process-Packet	1483	Toot-Contacts	1530
Tcp-Reply	1484	Toot-Equipped-Item	1531
Tcp-Socket-Input	1485	Toot-Fairy-Dust	1532
Tcp-Stream-Authenticate	1486	Toot-Has-Item-P	1533
Tcp-Unicast	1487	Toot-Info	1534
Template->Openapi	1488	Toot-Inventory	1537
Template-Match	1489	Toot-Join-Message	1538
Terrain	1490	Toot-Last-Active	1539
Terrain-Db-Key	1491	Toot-Last-Active, SetF	1539
Terrain-Exists-P	1492	Toot-List-Message	1540
Terrain-Height-Latitude	1494	Toot-Name	1541
Terrain-Height-Latitude, SetF	1494	Toot-Name, SetF	1541
Terrain-Height-Longitude	1495	Toot-Note	1542
Terrain-Height-Longitude, SetF	1495	Toot-Note, SetF	1542
Terrain-Height-P	1496	Toot-P	1543
Terrain-Height-Terrain	1497	Toot-Pad-Color	1544
Terrain-Height-Terrain, SetF	1497	Toot-Pad-Color, SetF	1544
Terrain-Height-World	1498	Toot-Pad-Color-Name-P	1546
Terrain-Height-World, SetF	1498	Toot-Passport-Stamped-P	1547
Terrain/ Add-Cactus	1499	Toot-Passport-Stamps	1548
Terrain/ Add-Flowers	1500	Toot-Pattern	1549
Terrain/ Add-Log	1501	Toot-Pattern, SetF	1549
Terrain/ Add-Mushrooms	1502	Toot-Pattern-Color	1550
Terrain/ Add-Shaddow-Bush	1503	Toot-Pattern-Color, SetF	1550
Terrain/ Add-Shaddow-Pit	1504	Toot-Pattern-Color-Name-P	1552
Terrain/ Add-Shaddow-Stalagmite	1505	Toot-Pattern-Name-P	1554
Terrain/ Add-Small-Pond	1506	Toot-Peanuts	1555
Terrain/ Add-Tree	1507	Toot-Player	1557
Terrain/ Connect-Streams	1508	Toot-Player, SetF	1557
Terrain/ Stream-Present-P	1509	Toot-Position	1558
Testcensor	217	Toot-Position, SetF	1558
Texi-Ref	1511	Toot-Presentation-Name	1559
This-Month	29	Toot-Private-Message	1560
This-Year	30	Toot-Quiesced-Altitude	1562
Three-Chars-In-A-Row-P	1512	Toot-Quiesced-Altitude, SetF	1562
Tick-Weather-Day	1513	Toot-Quiesced-Attribs	1563
Tick-Weather-Minute	1514	Toot-Quiesced-Attribs, SetF	1563
Toot	1515	Toot-Quiesced-D3	1564
Toot, SetF	1515	Toot-Quiesced-D3, SetF	1564
Toot-Avatar	1516	Toot-Quiesced-Data	1565
Toot-Avatar, SetF	1516	Toot-Quiesced-Emotion	1566
Toot-Avatar-Scale-X	1517	Toot-Quiesced-Emotion, SetF	1566
Toot-Avatar-Scale-X, SetF	1517	Toot-Quiesced-Latitude	1567
Toot-Avatar-Scale-Y	1518	Toot-Quiesced-Latitude, SetF	1567
Toot-Avatar-Scale-Y, SetF	1518	Toot-Quiesced-Longitude	1568
Toot-Avatar-Scale-Z	1519	Toot-Quiesced-Longitude, SetF	1568
Toot-Avatar-Scale-Z, SetF	1519	Toot-Quiesced-Observed	1569
Toot-Base-Color	1520	Toot-Quiesced-Observed, SetF	1569
Toot-Base-Color, SetF	1520	Toot-Quiesced-P	1570
Toot-Base-Color-Name-P	1522	Toot-Quiesced-Peer-Address	1571
Toot-Buddy-List	1523	Toot-Quiesced-Peer-Address, SetF	1571

Toot-Quiesced-Toot .....	1572
Toot-Quiesced-Toot, SetF .....	1572
Toot-Quiesced-World .....	1573
Toot-Quiesced-World, SetF .....	1573
Toot-Quiesced-Wtl .....	1574
Toot-Quiesced-Wtl, SetF .....	1574
Toot-Sms-Messages .....	1575
Toot-Speak .....	1576
Toot-Uuid .....	1577
Toot-Uuid, SetF .....	1577
Tootsville-V-Banner .....	1580
Trace-Log-File .....	1581
Trace-Output-Heartbeat .....	1582
Try-Reconnect-Toot-Name .....	1583
Two-Chars-In-A-Row-P .....	1584

## U

Un-Banhammer-Ip-Address .....	1586
Unbuild .....	218
Unicast .....	1587
Unimplemented-Feature .....	1590
Unimplemented-Feature, SetF .....	1590
Update-Gravatar .....	1592
Update-Toot-Last-Active .....	1594
Uptime .....	219
Uri-To-Uuid .....	1595
Url-To-String .....	1596
User->Alist .....	1597
User-Account .....	1598
User-Account, SetF .....	1598
User-Display-Name .....	1599
User-Email .....	1600
User-Face .....	1601
User-Given-Name .....	1602
User-Id .....	1603
User-Online-P .....	1604
User-Stream .....	1605
User-Surname .....	1606
Uuid-String-P .....	1607
Uuid-String-To-Base64 .....	1608
Uuid-To-Base64 .....	1609
Uuid-To-Uri .....	1610

## V

V .....	220
Valid-Child-Code-P .....	1611
Validate-Dns-Value .....	55
Value-To-Texi .....	1612
Vanish-Item .....	1613
Verbose-Log-File .....	1614
Verbosebugs .....	221
Version-Info-For .....	1615
Version-Info-List .....	1616
Version-Info-Report .....	1617
Version-Info-Report-String .....	1618

## W

Wall .....	222
Wallet-Info .....	1619
Wallops .....	223
Wallzones .....	224
Wants-Json-P .....	1620
Warning! .....	105
Wear-Slot-Alternate .....	1622
Wear-Slot-Alternate, SetF .....	1622
Wear-Slot-Avatar-Point .....	1623
Wear-Slot-Avatar-Point, SetF .....	1623
Wear-Slot-Id .....	1624
Wear-Slot-Id, SetF .....	1624
Wear-Slot-Info .....	1625
Wear-Slot-Name .....	1626
Wear-Slot-Name, SetF .....	1626
Wear-Slot-Obstruct-Max .....	1627
Wear-Slot-Obstruct-Max, SetF .....	1627
Wear-Slot-Obstruct-Min .....	1628
Wear-Slot-Obstruct-Min, SetF .....	1628
Wear-Slot-Obstruct-Point .....	1629
Wear-Slot-Obstruct-Point, SetF .....	1629
Wear-Slot-P .....	1630
Wear-Slot-Valence .....	1631
Wear-Slot-Valence, SetF .....	1631
Websocket-Authenticate .....	1633
Whatis .....	225
Whereami .....	226
Whereis .....	227
Which-Toot-Is-Not-Yours .....	1635
Which-Toot-Is-Not-Yours, SetF .....	1635
Whitespace-Char-P .....	1636
Whitespacep .....	1637
Who .....	228
Who-Is-Connected .....	1638
Whoami .....	229
Whoareyou .....	230
Wind-Vector-P .....	1640
Wind-Vector-X-Magnitude .....	1641
Wind-Vector-X-Magnitude, SetF .....	1641
Wind-Vector-Y-Magnitude .....	1642
Wind-Vector-Y-Magnitude, SetF .....	1642
Wind-X .....	1643
Wind-Y .....	1644
With-Cluster-Wide-Lock-Held .....	1645
With-Configuration .....	106
With-Continuable-Errors-Skipped .....	1646
With-Dbi .....	1647
With-Errors-As-Http .....	1648
With-Gather .....	1778
With-Http-Conditions .....	1649
With-Http-Errors-As-Infinity-Errors .....	1650
With-Local-Toot .....	1651
With-Local-User .....	1652
With-Maintenance-Times .....	1653
With-Memcached-Query .....	1654
With-Mulligan-Handlers .....	124
With-Pool-Thread-Restarts .....	125

With-Posted-Json.....	1655	Ws-Stats-Reset-All.....	1688
With-Rollbar-For-Debugger.....	107	Ws-To-Infinity.....	1689
With-Score-In-Range.....	1656	Ws-Unicast.....	1690
With-Standard-Streams-To-String.....	1657	Ws-Without-Login.....	1691
With-Twilio-Params.....	1779	Wtl-Course.....	1692
With-User.....	1658	Wtl-Course-Altitude.....	1693
With-Websocket-Disconnections.....	1659	Wtl-Course-Altitude, SetF.....	1693
Without-Medal.....	1660	Wtl-Course-End-Point.....	1694
Without-Sem.....	1661	Wtl-Course-End-Point, SetF.....	1694
World.....	1662	Wtl-Course-End-Time.....	1695
World-Mistp.....	1663	Wtl-Course-End-Time, SetF.....	1695
World-Moniker.....	1664	Wtl-Course-Facing.....	1696
World-Moniker, SetF.....	1664	Wtl-Course-Facing, SetF.....	1696
World-Moniker-P.....	1665	Wtl-Course-Latitude.....	1697
World-Name.....	1666	Wtl-Course-Latitude, SetF.....	1697
World-Name, SetF.....	1666	Wtl-Course-Longitude.....	1698
World-P.....	1667	Wtl-Course-Longitude, SetF.....	1698
Write-Class-Docs.....	1668	Wtl-Course-P.....	1699
Write-Docs.....	1669	Wtl-Course-Speed.....	1700
Write-Docs-Header.....	1670	Wtl-Course-Speed, SetF.....	1700
Write-Documentation.....	1671	Wtl-Course-Start-Point.....	1701
Write-Function-Docs.....	1672	Wtl-Course-Start-Point, SetF.....	1701
Write-Setf-Docs.....	1673	Wtl-Course-Start-Time.....	1702
Write-Staff-Journal-Entry.....	1674	Wtl-Course-Start-Time, SetF.....	1702
Ws-Approve-Toot.....	1675	Wtl-Course-World.....	1703
Ws-Bandwidth-By-Source.....	231, 1676	Wtl-Course-World, SetF.....	1703
Ws-Bandwidth-Record.....	1677	Www-Uri-Like-P.....	1705
Ws-Broadcast.....	1678		
Ws-Deny-Toot.....	1680	<b>Y</b>	
Ws-Evacuate-All.....	1681	Yesterday.....	1706
Ws-Kick.....	1682	Yield-Mariadb-Lock.....	1707
Ws-Kick-Other-Streams-For-User.....	1683		
Ws-Perform-Sign-In.....	1684	<b>Z</b>	
Ws-Reply.....	1685	Zoom.....	233
Ws-Sign-In-User.....	1686		
Ws-Stats.....	232, 1687		



## A.3 Variables

\*

*403.Json-Bytes*	250
*Acceptors*	251
*Access-Token*	58
*Api-Key*	32
*Application-Root*	252
*Async-Channel*	253
*Async-Tasks*	254
*Banhammer*	255
*Build-Date*	256
*Cassandra-Blacklist*	257
*Cassandra-Redlist*	258
*Client*	259
*Cluster*	260
*Code-Version*	59
*Compilation*	261
*Compiled*	262
*Config-File*	263
*Db*	264
*Dbi-Connection*	265
*Developmentp*	110
*Elevation-Map*	266
*Endpoint-List*	267
*Endpoints*	268
*Environment*	60
*Extensions-For-Content-Types*	287
*Framework*	61
*Google-Account-Keys-Refresh*	288
*Habitat-Map*	289
*Http-Status-Message*	290
*Humidity-Field*	292
*Ice-Credentials*	293
*Infinity-Ops*	294
*Infinity-Rest-Requests*	295
*Infinity-Stream-Requests*	296
*Infinity-Users*	297
*Infinity-Websocket-Resource*	298
*Maintenance-Tasks-Performed*	299
*Metronome*	300
*Metronome-Next-Tick*	301
*Metronome-Run*	302
*Metronome-Task-Lock*	303
*Metronome-Tasks*	304
*Motd*	305
*Mulligans*	111
*Npc-List*	306
*Person-Hook*	62
*Post-Tests-Queue*	307
*Robots*	308
*Running-Main-Loop*	309
*Server*	63
*Started*	310
*Tcp-Clients*	311
*Tcp-Listener*	312
*Tcp-Peer-Traffic*	313

*The-Metronome-Thread*	314
*Toot*	315
*Trace-Output-Heartbeat-Time*	316
*User*	317
*Utc-Timezone*	318
*Valid-Notifier-Levels*	64
*Verbose-Logging-Lock*	319
*Weather-Kernel*	320
*Websocket-Server*	321
*Wind-Vector-Field*	322
*Ws-Chars-Broadcast*	323
*Ws-Chars-Received*	324
*Ws-Chars-Unicast*	325
*Ws-Client-For-Toot*	326
*Ws-Client-For-User*	327
*Ws-Connections*	328
*Ws-Sign-Ins*	329
*Ws-Surprise-Disconnects*	330
*Ws-Traffic-Commands*	331
*Ws-Traffic-From*	332
*Ws-Traffic-Other*	333

+

+Alexa-Timestamp-Tolerance+	334
+Amazon-Cert-Chain-Url-Matching+	335
+Backtrace-Regex+	336
+Builder-Toot-Hard-Hat-Template+	337
+Color24-Names+	338
+Color24-Values+	339
+Context-Forms+	65
+Credits+	340
+Doc-Packages+	341
+Facing-Angles+	342
+Gravatar-Base-Uri+	343
+Habitat-Colors+	344
+Initial-T-Shirt-Colors+	345
+Max-Queue-Size-For-Thread-Pool+	112
+Moon-Day+	346
+Moon-Year+	347
+Other-Moon-Day+	348
+Other-Moon-Year+	349
+Pink-Moon-Day+	350
+Pink-Moon-Year+	351
+Pre-Login-Max-Commands+	352
+Pre-Login-Max-Time+	353
+Single-Core-Threads+	113
+Supported-Languages+	354
+Threads-Per-Core+	114
+Toot-Base-Color-Names+	355
+Toot-Basic-Pattern-Names+	356
+Toot-Extended-Pattern-Names+	357
+Toot-Pad-Color-Names+	358
+Toot-Pattern-Color-Names+	359
+Unix-Time-In-Universal+	360
+Unix-Zero-In-Universal-Time+	361

+Ws-Idle-Seconds+ ..... 362

**D**

Devel..... 626

**P**

Prod..... 1301

Pull-Records-Cache ..... 1304

**Q**

Qa..... 1305

**T**

Test..... 1510

## A.4 Data types

### A

Avatar ..... 389  
Avatar-Slot ..... 397

### B

Bad-Request ..... 406  
Basic-8-Personality ..... 417

### C

Chaos-Personality ..... 436  
Character-Music ..... 437  
Child-Code ..... 453  
Child-Request ..... 454  
Cluster-Wide-Lock-Busy-Error ..... 471  
Cluster-Wide-Lock-Busy-Warning ..... 472  
Cluster-Wide-Lock-Condition ..... 473  
Cluster-Wide-Lock-Error ..... 474  
Cluster-Wide-Lock-Not-Locked ..... 475  
Cluster-Wide-Lock-Not-Ours ..... 476  
Cname-Already-On-Record ..... 33  
Cname-Must-Be-Only-Record ..... 34  
Color24 ..... 477  
Contact ..... 511  
Credential ..... 557  
Cupid-Personality ..... 567

### D

Dns-Name ..... 632  
Doodle-Personality ..... 643  
Dottie-Personality ..... 644  
Dreamhost-API-Error ..... 38  
Dreamhost-API-Error-With-Details ..... 39  
Dreamhost-API-Warning ..... 40

### E

Endpoint ..... 653

### F

Flora-Personality ..... 848

### G

Game-Point ..... 874  
Gone ..... 899  
Gossip-Initiation ..... 900

### H

Harmony-Personality ..... 919  
Holiday-Special-Personality ..... 921  
Http-Client-Error ..... 925  
Http-Error ..... 80  
Http-Idempotent-Request-Method ..... 926  
Http-Request-Method ..... 928  
Http-Safe-Request-Method ..... 929

### I

Infinity-Websocket-Resource ..... 1027  
Internal-Error-Could-Not-Add-Record ..... 42  
Internal-Error-Could-Not-Load-Zone ..... 43  
Internal-Error-Updating-Zone ..... 44  
Invalid-Record ..... 45  
Invalid-Type ..... 46  
Invalid-Value ..... 47  
Inventory-Item ..... 1037  
Item ..... 1044  
Item-Template ..... 1061

### J

Jack-Personality ..... 1085

### K

Kind-Of-Habitat ..... 1090

### L

Lil-Mc-Personality ..... 1095  
Locale-Music ..... 1105  
Login ..... 1112  
Lot ..... 1127

### M

Map-Places ..... 1176  
Mayor-Louis-Personality ..... 1178  
Metronome-Task ..... 1183  
Mist ..... 1190  
Moo-Personality ..... 1200  
Music ..... 1202

**N**

Nevermind-Personality .....	1213
No-Record .....	48
No-Such-Zone .....	49
No-Type .....	50
No-Value .....	51
Not-Found .....	1215
Not-Your-Toot-Error .....	1218

**P**

Parent-Child .....	1223
Pattern .....	1235
Person .....	1242
Person-Link .....	1253
Picasso-Personality .....	1266
Place .....	1267
Props-Personality .....	1302

**Q**

Quaestor-Event .....	1309
----------------------	------

**R**

Rad-Personality .....	1329
Record-Already-Exists-Not-Editable .....	52
Record-Already-Exists-Remove-First .....	53
Robot .....	1357
Robot-Chaos .....	1359
Robot-Cupid .....	1362
Robot-Doodle .....	1363
Robot-Dottie .....	1364
Robot-Flora .....	1365
Robot-Harmony .....	1368
Robot-Jack .....	1371
Robot-Lil-Mc .....	1372
Robot-Mayor-Louis .....	1375
Robot-Moo .....	1377
Robot-Nevermind .....	1378
Robot-Picasso .....	1379
Robot-Props .....	1381
Robot-Rad .....	1382
Robot-Shade .....	1385
Robot-Smudge .....	1386
Robot-Snowcone .....	1387
Robot-Sparkle .....	1388
Robot-Splot .....	1389
Robot-Superstar .....	1390
Robot-Zap .....	1392

**S**

Shaddow-Personality .....	1417
Shade-Personality .....	1418
Sms .....	1425
Smudge-Personality .....	1433
Snowcone-Personality .....	1434
Sparkle-Personality .....	1435
Splot-Personality .....	1439
Store-Item .....	1457
Superstar-Personality .....	1469

**T**

Tcp-Client .....	1475
Terrain-Height .....	1493
Thread-Pool-Taskmaster .....	123
Toot .....	1515
Toot-Base-Color-Name .....	1521
Toot-Name .....	1541
Toot-Pad-Color-Name .....	1545
Toot-Pattern-Color-Name .....	1551
Toot-Pattern-Name .....	1553
Toot-Personality .....	1556
Toot-Quiesced .....	1561
Tootsville-Rest-Acceptor .....	1578
Tootsville-Rest-Ssl-Acceptor .....	1579
Two-Letter-String .....	1585

**U**

Unidentified-Player-Error .....	1588
Unimplemented .....	1589
Unprocessable .....	1591
Update-Nil .....	1593

**W**

Wear-Slot .....	1621
Websocket-Acceptor .....	1632
Websocket-Ssl-Acceptor .....	1634
Wind-Vector .....	1639
World .....	1662
World-Moniker .....	1664
Ws-Client .....	1679
Wtl-Course .....	1692
Www-Uri .....	1704

**Z**

Zap-Personality .....	1708
-----------------------	------

## A.5 Pathnames

### A

acceptor.lisp . . . . . 366, 629, 842, 904, 905, 906, 917,  
918, 1216, 1352, 1353, 1406, 1466, 1489, 1636, 1637,  
1649, 1709, 1710, 1711, 1712, 1713, 1714, 1715, 1716  
alexa.lisp . . . . . 442, 443, 444, 445, 447, 448, 452, 579,  
586, 818, 1214, 1415, 1736, 1737, 1738, 1739, 1740,  
1741, 1742, 1743, 1744, 1745, 1746, 1747, 1748  
auth-firebase.lisp . . . . . 415, 426, 449, 498, 816, 892,  
927, 1222, 1467, 1717, 1760

### B

binary.lisp . . . . . 425, 799, 803, 1033, 1416, 1595, 1610  
browser.lisp . . . . . 820

### C

cassandra.lisp . . . . . 429, 430, 431, 432, 433, 434, 435  
Chcerogryllum.lisp . . . . . 14, 15, 16, 17, 18, 19, 20, 21,  
22, 23, 24, 25, 27, 28, 29, 30  
characters.lisp . . . . . 587, 590, 1032  
chat.lisp . . . . . 705  
clock.lisp . . . . . 706  
color+pattern.lisp . . . . . 450, 478, 479, 480,  
481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 524,  
1034, 1142, 1230, 1356  
command-line.lisp . . . . . 810, 1299  
config.lisp . . . . . 383, 468, 469, 470, 502, 582, 651, 1102,  
1126, 1441, 1442  
contacts.lisp . . . . . 371, 615, 1530

### D

date+time.lisp . . . . . 363, 364, 896, 920, 1094, 1706  
db-central.lisp . . . . . 385, 416, 491, 492, 493, 494, 495,  
596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606,  
607, 608, 609, 610, 611, 612, 613, 614, 636, 804, 893,  
935, 1097, 1303, 1608, 1609  
dreamhost.lisp . . . . . 35, 36, 37, 54, 55  
dumper-2SKVI5f7.lisp . . . . . 72, 73, 75, 87, 105

### E

endpoint.lisp . . . . . 466, 657, 658, 701, 702, 792, 793,  
795, 823, 825, 827, 1232, 1345  
errors.lisp . . . . . 500, 501, 1229, 1423, 1437

### F

friendly.lisp . . . . . 390, 391, 392,  
394, 395, 396, 398, 399, 400, 401, 402, 438, 439, 440,  
455, 456, 458, 459, 460, 461, 462, 463, 512, 513, 514,  
515, 516, 517, 518, 520, 521, 522, 523, 525, 526, 529,  
530, 531, 532, 533, 534, 536, 537, 538, 539, 540, 541,  
542, 543, 544, 545, 549, 550, 551, 552, 554, 558, 559,  
560, 561, 562, 563, 564, 565, 566, 1038, 1040, 1041,  
1042, 1043, 1045, 1046, 1047, 1048, 1049, 1050, 1051,  
1052, 1056, 1057, 1060, 1061, 1062, 1063, 1064, 1065,  
1066, 1067, 1068, 1069, 1070, 1071, 1073, 1074, 1075,  
1076, 1077, 1078, 1079, 1080, 1081, 1082, 1083, 1106,  
1107, 1108, 1109, 1110, 1111, 1114, 1117, 1119, 1120,  
1121, 1122, 1123, 1124, 1128, 1129, 1130, 1131, 1132,  
1133, 1134, 1135, 1136, 1137, 1138, 1139, 1140, 1141,  
1143, 1144, 1148, 1149, 1150, 1151, 1152, 1153, 1155,  
1156, 1158, 1159, 1160, 1161, 1162, 1163, 1165, 1166,  
1168, 1169, 1170, 1171, 1174, 1191, 1192, 1193, 1194,  
1195, 1196, 1197, 1198, 1199, 1203, 1204, 1205, 1206,  
1207, 1208, 1209, 1224, 1225, 1226, 1236, 1237, 1238,  
1243, 1245, 1246, 1248, 1249, 1252, 1254, 1255, 1256,  
1257, 1258, 1259, 1260, 1262, 1263, 1264, 1265, 1268,  
1269, 1270, 1272, 1273, 1274, 1275, 1276, 1279, 1280,  
1310, 1311, 1312, 1313, 1314, 1315, 1316, 1317, 1318,  
1319, 1320, 1321, 1426, 1427, 1429, 1430, 1431, 1432,  
1458, 1459, 1460, 1461, 1462, 1463, 1494, 1495, 1496,  
1497, 1498, 1516, 1517, 1518, 1519, 1520, 1527, 1539,  
1541, 1542, 1543, 1544, 1549, 1550, 1557, 1562, 1563,  
1564, 1566, 1567, 1568, 1569, 1570, 1571, 1572, 1573,  
1574, 1577, 1622, 1623, 1624, 1626, 1627, 1628, 1629,  
1630, 1631, 1664, 1666, 1667

### G

game-actions.lisp . . . . . 853, 854, 855, 856, 857, 858,  
859, 860, 861, 862, 863, 865, 866, 867, 868, 869, 870,  
871, 872, 873  
generic-db.lisp . . . . . 570, 577, 623, 832, 833, 834, 835,  
933, 1036, 1103, 1164, 1400  
gossip.lisp . . . . . 464, 528, 819, 901, 902, 903, 931, 932,  
1147

### H

http-error.lisp . . . . . 847, 1761  
http-types.lisp . . . . . 1298

### I

infinity.lisp . . . . . 403, 427, 428, 589, 594, 787, 1023, 1650  
info.lisp . . . . . 707  
items.lisp . . . . . 393, 556, 639, 640, 641, 642, 646, 897,  
907, 1053, 1055, 1058, 1059, 1072, 1277, 1278, 1456,  
1472, 1533, 1537, 1613, 1625

**L**

legacy-commands.lisp . . . . . 365, 647, 724,  
726, 727, 729, 731, 733, 734, 735, 737, 738, 739, 740,  
741, 742, 744, 746, 748, 749, 750, 751, 752, 753, 754,  
755, 756, 757, 758, 759, 760, 761, 763, 766, 767, 768,  
769, 771, 772, 773, 774, 775, 776, 778, 779, 781, 783,  
852, 885, 936, 938, 940, 942, 945, 947, 948, 951, 953,  
955, 959, 960, 961, 963, 966, 968, 973, 974, 975, 976,  
977, 978, 979, 980, 981, 982, 985, 986, 987, 988, 993,  
996, 1001, 1002, 1003, 1006, 1007, 1008, 1010, 1011,  
1013, 1015, 1018, 1022, 1025, 1231, 1271, 1346, 1412,  
1413, 1414, 1523  
legacy-ops.lisp . . . 128, 130, 131, 132, 133, 134, 136,  
137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 150,  
151, 152, 153, 154, 155, 156, 157, 158, 159, 161, 162,  
163, 164, 165, 166, 167, 168, 169, 171, 172, 173, 174,  
176, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187,  
188, 192, 193, 196, 197, 198, 199, 200, 201, 202, 205,  
206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216,  
217, 218, 220, 221, 222, 223, 224, 225, 226, 227, 228,  
229, 230, 233, 236, 237, 238, 239, 240, 241, 242, 243,  
244, 245, 246, 247, 248, 249, 408, 1087, 1098, 1586  
logging.lisp . . . . . 409, 410, 411, 412, 413,  
414, 813, 828, 910, 911, 912, 913, 1221, 1407, 1408,  
1409, 1410, 1411, 1444, 1581, 1582, 1614

**M**

main.lisp . . . 376, 405, 504, 578, 617, 619, 620, 621,  
822, 1031, 1211, 1338, 1342, 1344, 1398, 1445, 1447,  
1448, 1450, 1452, 1455, 1470  
maria.lisp . . . 421, 422, 505, 571, 572, 573, 574, 575,  
576, 634, 894, 1645, 1647, 1707  
memcached.lisp . . 503, 812, 1179, 1295, 1327, 1654,  
1734, 1735  
memoization.lisp . . 115, 379, 380, 1292, 1522, 1546,  
1552, 1554  
messaging.lisp . . . . . 420, 838, 1587  
metronome.lisp . . . . . 535, 633,  
635, 1154, 1180, 1181, 1182, 1184, 1185, 1186, 1187,  
1188, 1189, 1341, 1399, 1446, 1453  
modern-ops.lisp . . 129, 135, 147, 148, 149, 160, 170,  
175, 194, 195, 203, 204, 219, 231, 232, 1086

**N**

new-commands-20.lisp . . . . 424, 728, 736, 747, 762,  
764, 765, 777, 782, 784, 785, 786, 941, 952, 972, 989,  
994, 995, 1014, 1024, 1026, 1029, 1030, 1035, 1104,  
1157, 1281, 1282, 1288, 1331, 1422, 1540, 1594

**P**

package-post.lisp . . . . . 1720  
passport.lisp . . . . . 840, 1443, 1547, 1548  
power-on-self-test.lisp . . . . . 595, 1291, 1294

**Q**

quaestor.lisp . . 496, 497, 850, 851, 1306, 1307, 1308,  
1322, 1323, 1324, 1449, 1524, 1532, 1555, 1656, 1660

**R**

redirect.lisp . . . . . 1339, 1340  
robo-toot.lisp . . . . . 1367, 1370, 1373, 1391  
robots.lisp . . . . . 381, 527, 555, 591, 837, 875,  
876, 877, 878, 879, 880, 881, 882, 1093, 1101, 1125,  
1146, 1175, 1212, 1233, 1343, 1354, 1358, 1361, 1374,  
1380, 1384, 1393, 1565, 1662, 1692, 1693, 1694, 1695,  
1696, 1697, 1698, 1699, 1700, 1701, 1702, 1703  
rollbar.lisp . . . . 66, 67, 68, 69, 70, 71, 74, 76, 77, 78,  
79, 86, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99,  
100, 101, 102, 103, 104, 106, 107

**S**

slash-gossip.lisp . . . . . 665, 666, 667, 708, 709  
slash-login.lisp . . . . . 715  
slash-maintenance.lisp . . . . 588, 668, 716, 717, 718,  
719, 720, 721, 1646, 1653, 1657  
slash-meta-game.lisp . . 499, 580, 581, 637, 638, 654,  
655, 656, 669, 670, 671, 672, 673, 794, 796, 797, 798,  
811, 836, 844, 914, 1234, 1350, 1488, 1728  
slash-toots.lisp . . . . . 674, 675, 722, 788  
slash-users.lisp . . . 660, 676, 677, 678, 679, 680, 681,  
704, 723, 789, 1287  
slash-version.lisp . . . . . 682, 684, 685, 686  
slash-world.lisp . . . 625, 687, 688, 689, 690, 691, 692,  
693, 694, 695, 696, 697, 698, 699  
sms.lisp . . . . . 1404, 1428, 1575  
staff-journal.lisp . . . . . 801, 1333, 1334, 1674  
string-characteristics.lisp . . . 846, 1096, 1464, 1465,  
1512, 1584, 1607  
sun-moon.lisp . . . . . 1201, 1468

**T**

tcp-stream.lisp . . . . . 546,  
824, 1167, 1405, 1451, 1473, 1474, 1476, 1477, 1478,  
1479, 1480, 1481, 1482, 1483, 1484, 1485, 1486, 1487  
terrain.lisp . . . . . 547, 548, 648, 821, 831, 884, 887,  
888, 889, 890, 891, 898, 915, 916, 1289, 1419, 1424,  
1436, 1490, 1491, 1492, 1499, 1500, 1501, 1502, 1503,  
1504, 1505, 1506, 1507, 1508, 1509, 1718, 1719  
thread-pool-taskmaster.lisp . . . . 116, 117, 118, 119,  
120, 124, 125  
toot-names.lisp . . . . . 451, 1293, 1347, 1611  
toots.lisp . . . . . 616, 624, 806, 814, 839, 1039, 1328,  
1348, 1525, 1526, 1528, 1529, 1531, 1536, 1559, 1619  
tootsville-commands.lisp . . . 725, 730, 732, 743, 745,  
770, 780, 937, 943, 944, 965, 967, 1005, 1019  
twilio-simple.lisp . . . . 1764, 1765, 1766, 1767, 1768,  
1769, 1770, 1771, 1772, 1773, 1774, 1775, 1776, 1777,  
1778, 1779

twilio.lisp . . . . . 710, 711, 712, 713, 714

## U

uri-types.lisp . . . . . 922, 923, 1705, 1725, 1726, 1727,  
1729, 1730, 1731, 1732, 1733, 1749, 1750, 1751, 1752,  
1753, 1754, 1755, 1756, 1757, 1758, 1759  
users.lisp . . . . . 377, 382, 386,  
387, 423, 457, 649, 807, 829, 830, 841, 895, 908, 909,  
934, 1099, 1113, 1227, 1228, 1240, 1241, 1244, 1247,  
1250, 1251, 1261, 1283, 1284, 1285, 1286, 1290, 1335,  
1336, 1337, 1401, 1402, 1592, 1596, 1597, 1599, 1600,  
1601, 1602, 1603, 1606, 1651, 1652, 1658  
utils.lisp . . 419, 441, 627, 631, 650, 1420, 1438, 1471

## V

version.lisp . . 805, 817, 1219, 1394, 1395, 1396, 1397,  
1580, 1615, 1616, 1617, 1618, 1721, 1722, 1723, 1724  
view.lisp . . . . . 1349

## W

weather.lisp . . . 467, 553, 569, 809, 886, 1172, 1173,  
1297, 1513, 1514, 1640, 1641, 1642, 1643, 1644  
web.lisp . . . . . 368, 370, 374, 384, 388, 446, 510,  
519, 583, 584, 585, 622, 652, 662, 663, 664, 700, 815,  
845, 1091, 1145, 1210, 1325, 1326, 1332, 1351, 1403,  
1620, 1648, 1655, 1661  
websockets.lisp . . 369, 373, 375, 404, 507, 508, 509,  
628, 802, 826, 843, 849, 958, 984, 990, 1088, 1089,  
1100, 1115, 1116, 1118, 1177, 1300, 1355, 1454, 1515,  
1538, 1558, 1560, 1576, 1583, 1604, 1605, 1633, 1638,  
1659, 1675, 1676, 1677, 1678, 1680, 1681, 1682, 1683,  
1684, 1685, 1686, 1687, 1688, 1689, 1690, 1691  
world-types.lisp . . . . . 1665  
world.lisp . . . 618, 630, 1054, 1084, 1421, 1440, 1663  
write-docs-2.lisp . . . 378, 465, 645, 883, 1511, 1612,  
1668, 1669, 1670, 1671, 1672, 1673

## A.6 Infinity Mode commands

### A

addFurniture: Alias for  
INFINITY-SET-FURNITURE. . . . . 936  
addJournalEntry: Add a staff journal entry. . . . . 937  
addToList: Add a user to a buddy list or ignore  
list (removed in 1.2) . . . . . 938

### C

click: Used by the client to report a mouse  
click or finger tap. . . . . 939  
considerChildApproval: Consider whether to  
approve a child's request with ID UUID. . . . . 941  
createUserHouse: Either claim the user's house and  
lot, or add a room to their house. . . . . 942

### D

deleteMailMessage: Delete a message from the  
user's (SMS) mailbox. . . . . 943  
doff: Remove clothes or Pivitz. . . . . 944  
dofff: Doff all clothing items. . . . . 945  
don: Don (or equip) an item . . . . . 946

### E

echo: Echoes back the supplied JSON (or  
ActionScript) object to the client. . . . . 948  
endEvent: Attempt to end an event. . . . . 949  
enumerateWearSlots: Enumerates all possible  
wear slots for any avatar. . . . . 952

### F

finger: Get public info for a list of Toots. . . . . 953

### G

gameAction: Send an in-world game's action. . . . . 954  
getAvatars: Get avatar data for a list  
of (other) users. . . . . 959  
getColorPalettes: getColorPalettes . . . . . 960  
getInventory: Get all inventory for an user  
(themselves) — both active and inactive . . . . . 961  
getInventoryByType: Get a subset of items  
from your own inventory. . . . . 962  
getMailInBox: Get a listing of messages in  
an SMS mailbox. . . . . 964  
getOnlineUsers: Get a list of users online. . . . . 966  
getPassport: Get the list of places that the user  
has gotten a passport stamp at. . . . . 967  
getRoomList: Get a list of all "well known" Rooms  
currently active/visible. . . . . 968  
getRoomVars: Returns "room variables." . . . . . 969

getServerTime: Send the server time to the  
client requesting it. . . . . 973  
getSessionApple: Initialise a session key for stream  
or batch mode operations. . . . . 974  
getStoreItemInfo: Get information about items in a  
store which can be purchased. . . . . 975  
getUserLists: Get the user's buddy  
list and ignore list. . . . . 976  
getWallet: Get the contents of the player's wallet  
(peanuts and fairy dust) . . . . . 977  
getZoneList: Get a list of all Zones  
currently active/visible. . . . . 978  
give: Give an item to another user. . . . . 979  
go: go to a place and/or perform a gesture . . . . . 980

### I

initUserRoom: Create a user's private  
room (in their house). . . . . 981

### J

join: Join a room. . . . . 982

### L

logout: Log out of this game session . . . . . 985

### M

mailCustomerService: Send an eMail to  
customer service (feedback) . . . . . 986

### P

peekAtInventory: Look at other  
users' inventories . . . . . 987  
ping: Send a ping to the server to  
get back a pong. . . . . 988  
playWith: Choose a Toot as your active  
CHARACTER in the game. . . . . 989  
promptReply: Accept a reply to a  
server-initiated prompt . . . . . 991

### Q

quiesce: Quiesce Toot values to database for  
logout, or periodically as a backup. . . . . 994



**R**

- readMap: Get the positions of badges and named locations on the map. . . . . 995
- removeFromList: Remove someone from a buddy list or ignore list. . . . . 996
- reportBug: This method allows the client to “phone home” to report a bug. . . . . 997
- reportUser: Report an user to the moderator(s) on duty for breaking a rule . . . . . 1002
- requestBuddy: Request adding a user to your buddy list (mutual-add) using the notification-based system . . . . . 1003

**S**

- sendMailMessage: Send an in-game SMS message. . . . . 1004
- sendOutOfBandMessage: Send an arbitrary JSON packet to another user, or all of the users . . 1006
- serverTime: Accept the client’s notification of a server-time adjustment. . . . . 1007
- setAvatarColor: Set the avatar base and extra (pad) colours for the given user. . . . . 1008
- setFurniture: Set or change a furniture item. . . . . 1009
- setRoomVar: Set a room variable or set of room variables. . . . . 1011

- setUserVar: Set “User Variables” . . . . . 1012
- shoot: Fire a shot from a projectile device. . . 1014
- spawnZone: Spawn an additional server peer pairing. . . . . 1015
- speak: The user speaks SPEECH at volume VOL in public. . . . . 1016
- stampPassport: Stamp the Toot’s passport . . 1019
- startEvent: Attempt to begin a Quaestor Event. Might return an error. . . . . 1020

**T**

- tootList: Enumerates all Toots owned by the user. . . . . 1024

**U**

- useEquipment: The player wishes to use a piece of equipment on a particular item or place. . . 1025

**W**

- wardrobe: Describe what your Toot is wearing. . . . . 1026
- wtl: Walk the Line. . . . . 1028
- wtl4: Walk the Line indirect refresher from observer . . . . . 1030

## A.7 Operator commands

#\$: Execute a command script. . . . .	128	#getmotd: Retrieve the current Message Of The Day as a server message. . . . .	163
#addevent: Add a GameEvent to a Zone. . . . .	132	#getschedule: Get schedule. . . . .	164
#agent: Set the clothing and colors of a robot to match the invoking user. . . . .	133	#getschedulefor: Get scheduled events for a particular class (scheduled by that class) . . . . .	165
#apropos: Runs APROPOS (see the Common Lisp HyperSpec) for a remote user. . . . .	129	#getuvar: Get a user variable. . . . .	166
#askme: Used to test the question-and-answer subsystem. . . . .	134	#getuvars: Get all user variables for a given user. . . . .	167
#at: Issue an operator command on a particular server instance. . . . .	135	#getvar: Get a room variable. . . . .	168
#ban: Ban a user persistently (permanently) from the game. . . . .	136	#getvars: Get all room variables. . . . .	169
#banhammer: Ban an IP address from connecting. . . . .	137	#git-pull: Does a <b>git pull</b> in the server directory. . . . .	170
#beam: Beam yourself to a different location. . . . .	138	#give: Give an item as a gift to another user. . . . .	171
#census: Load a number of users. . . . .	139	#givehead: Grants a new inventory item to a user and equips it. . . . .	172
#clearbadge: Clear a badge off the map. . . . .	140	#grant: Grants a new inventory item to a user. . . . .	173
#clearcache: Forcibly clear all cachés (MemCacheD). . . . .	141	#headcount: Get headcount information about the running system. . . . .	174
#clear event: Clear a GameEvent. . . . .	142	#infinity-stats: Returns some statistics about Infinity-mode requests. . . . .	175
#clearvar: Clear a room variable. (no longer supported). . . . .	143	#inv: Get a user's inventory. . . . .	176
#cloneroom: Clone a room. (no longer supported). . . . .	144	#journal: Add a staff journal entry or review last entries. . . . .	1086
#createroom: Create a new room. . . . .	145	#kick: Kick a user offline for a certain reason. . . . .	177
#dbcinfo: Get information from the DBI (database) layer. . . . .	146	#king: Upgrade a user account. . . . .	179
#doc: Obtain documentation string in raw form about a symbol. . . . .	147	#liftban: Lift the ban upon a user. . . . .	180
#doodle-pattern: Change the pattern of a Toot. . . . .	149	#loadlists: Reload the censorship lists. . . . .	181
#doodle: Change the colors of a Toot. . . . .	148	#mem: Display some memory usage and other debugging type information as a pop-up message. . . . .	182
#dress: Force a character to wear a specific clothing item. . . . .	150	#metronome: Display information about or micromanage the metronome. . . . .	183
#drop: Drop an item. . . . .	151	#motd: Set the message of the day. . . . .	184
#dropkick: Silently disconnect a user. . . . .	152	#mute: Mute a user or area. . . . .	185
#dumpthreads: Dump debugging information including all running threads to the server logs. . . . .	153	#nuke: Forcibly disconnect everyone in an area. . . . .	186
#enablepathfinder: Temporary test routine for testing pathfinders on users. . . . .	154	#parentapproves: Signal that a parent approves a user signing in. . . . .	187
#evacuate: Evacuate all users from the current Zone to another Zone. . . . .	155	#ping: Ping the server, to force a neutral administrative message reply. . . . .	188
#filter: Test censorship rules against words or phrases. . . . .	156	#place: Put a thing or a Place into the game. . . . .	189
#finger: Finger a user account. . . . .	157	#purgephysics: Purge pending physics interactions. . . . .	193
#flush: Historically, this flushed the database write caché. . . . .	158	#push-script: Instruct clients to load a new script file. . . . .	194
#game: Send a command to the operator command interpreter for a running game. . . . .	159	#quick-reload: Quicklisp reload of the Tootsville package from disk. . . . .	195
#gc: Perform immediate garbage collection. . . . .	160	#rc: Run an RC (Run Commands) script. . . . .	196
#getconfig: Reads a configuration key. . . . .	161	#reboot: Restart the game server. . . . .	197
#getevents: List GameEvents in your current Zone. . . . .	162	#reloadconfig: Reloads configuration properties. . . . .	198

#retire: Retire a server.....	199	#time: Displays a message with the current server time.....	130
#run: Run an arbitrary nullary Lisp function or method.....	200	#unbuild: Destroy a named spot.....	218
#saveroomvars: Save room variables.....	201	#uptime: Gives the uptime of the server software.....	219
#scotty: Force a user to relocate to another location.....	202	#v: Forces a user to say a message.....	220
#script: Push a new function into the TOOTSVILLE-USER package.....	203	#verbosebugs: Set verbose bug backtrace reporting on or off.....	221
#server-list: Enumerate the servers active in this cluster.....	204	#wall: Write to all players.....	222
#setavatarcolors: Sets the base and extra colors of a user's avatar.....	205	#wallops: Write to all operators.....	223
#setbadge: Set the badge on a map area.....	206	#wallzones: Write to all zones.....	224
#setconfig: Set a config property.....	207	#warn: Warn a user about breaking a rule....	131
#setstafflevel: Set the staff level for a user....	208	#whatis: Displays information about an item template.....	225
#setuvar: Set a user variable.....	209	#whereami: Return an administrative message with the name of the server to which.....	226
#setvar: Set a room variable.....	210	#whereis: Locate a user in the game world....	227
#shanghai: Force a client into a different room and zone.....	211	#who: Displays a list of everyone currently in a room.....	228
#shout: Speak in another zone.....	212	#whoami: Cause the character to speak his/her name in the current room.....	229
#spawnroom: Mark a "spot" in the game.....	213	#whoareyou: Ask the server who it is.....	230
#spawnzone: Create a new zone.....	214	#ws-bandwidth-by-source: Returns some statistics about WebSockets bandwidth by source....	231
#speak: Allows a user to speak who had previously been muted.....	215	#ws-stats: Returns some statistics about WebSockets connections.....	232
#stfu: Silences (mutes) a user.....	216	#zoom: Set the visual Zoom level of a room...	233
#testcensor: Test a message with the censor, displays the filter result.....	217		

## A.8 Game Actions

### B

bowlingStrikePins ..... 854

### C

cardGameArrange ..... 855  
 cardGameDeal ..... 856  
 cardGameDraw ..... 857  
 cardGameMove ..... 858  
 cardGamePlay ..... 859  
 cardGameShuffle ..... 860  
 cardGameTake ..... 861

### G

getBowlingScorecard ..... 862

### J

joinBowlingGame ..... 863  
 joinCardGame ..... 864

### O

Overview of Game Actions ..... 954

### P

partBowlingGame ..... 866  
 partCardGame ..... 867  
 pauseSportsBallTimer ..... 868

### R

resetBowlingPins ..... 853

### S

sportsBallGoal ..... 869  
 startBowling ..... 870  
 startSportsBallGame ..... 871  
 startSportsBallTimer ..... 872

### T

tagYouReIt ..... 873

## A.9 Javascript

### A

AvatarBuilder. addNameTag .....	1782
AvatarBuilder. build .....	1783
AvatarBuilder. colorize .....	1784
AvatarBuilder. enableShadows .....	1785
AvatarBuilder. getPathForPattern .....	1786
AvatarBuilder. loadAvatarBase .....	1787
AvatarBuilder. patterns .....	1788
AvatarBuilder. postBuild .....	1789
AvatarBuilder. rainbowColor .....	1790
AvatarBuilder. rememberAvatar .....	1791
AvatarBuilder. update .....	1792
AvatarViewer. createCamera .....	1793
AvatarViewer. createLight .....	1794
AvatarViewer. createScene .....	1795
AvatarViewer. createViewerInCanvas .....	1796
AvatarViewer. createViewerReally .....	1797
AvatarViewer. getAvatar .....	1798

### D

decodeTime .....	2269
------------------	------

### F

FurnitureBuilder. build .....	1799
FurnitureBuilder. build2 .....	1800
FurnitureBuilder. colorize .....	1801
FurnitureBuilder. enableShadows .....	1802
FurnitureBuilder. loadItemTemplate .....	1803
FurnitureBuilder. rememberItem .....	1804
FurnitureBuilder. update .....	1805

### G

Game. BallSystem. fastForward .....	1806
Game. BallSystem. register .....	1807
Game. BallSystem. remove .....	1808
Game. BallSystem. updateBalls .....	1809
Game. bootstrap .....	2004
Game. clickedOnItem .....	2005
Game. Commands. addFurniture .....	1810
Game. Commands. addToList .....	1811
Game. Commands. click .....	1812
Game. Commands. createUserHouse .....	1813
Game. Commands. doff .....	1814
Game. Commands. don .....	1815
Game. Commands. echo .....	1816
Game. Commands. endevent .....	1818
Game. Commands. endEvent .....	1817
Game. Commands. finger .....	1819
Game. Commands. gameAction .....	1820
Game. Commands. getApple .....	1821
Game. Commands. getAvatars .....	1822
Game. Commands. getColorPalettes .....	1823

Game. Commands. getInventory .....	1824
Game. Commands. getInventoryByType .....	1825
Game. Commands. getOnlineUsers .....	1826
Game. Commands. getRoomList .....	1827
Game. Commands. getServerTime .....	1828
Game. Commands. getSessionApple .....	1829
Game. Commands. getStoreItemInfo .....	1830
Game. Commands. getUserLists .....	1831
Game. Commands. getWallet .....	1832
Game. Commands. getZoneList .....	1833
Game. Commands. give .....	1834
Game. Commands. go .....	1835
Game. Commands. initUserRoom .....	1836
Game. Commands. join .....	1837
Game. Commands. login .....	1838
Game. Commands. logout .....	1839
Game. Commands. mailCustomerService .....	1840
Game. Commands. peekAtInventory .....	1841
Game. Commands. ping .....	1842
Game. Commands. promptReply .....	1843
Game. Commands. removeFromList .....	1844
Game. Commands. reportBug .....	1845
Game. Commands. reportUser .....	1846
Game. Commands. requestBuddy .....	1847
Game. Commands. sendOutOfBandMessage .....	1848
Game. Commands. serverTime .....	1849
Game. Commands. setAvatarColor .....	1850
Game. Commands. setFurniture .....	1851
Game. Commands. spawnZone .....	1852
Game. Commands. speak .....	1853
Game. Commands. startEvent .....	1854
Game. Commands. useEquipment .....	1855
Game. Commands. walk .....	1856
Game. credits .....	2006
Game. fastForward .....	2007
Game. Gatekeeper. admin .....	1857
Game. Gatekeeper. avatars .....	1858
Game. Gatekeeper. ayt .....	1859
Game. Gatekeeper. badgeUpdate .....	1860
Game. Gatekeeper. beam .....	1861
Game. Gatekeeper. bots .....	1862
Game. Gatekeeper. buddyList .....	1863
Game. Gatekeeper. buddyRequest .....	1864
Game. Gatekeeper. burgeon .....	1865
Game. Gatekeeper. bye .....	1866
Game. Gatekeeper. c .....	1867
Game. Gatekeeper. earning .....	1868
Game. Gatekeeper. endEvent .....	1869
Game. Gatekeeper. forceMove .....	1870
Game. Gatekeeper. gameAction .....	1871
Game. Gatekeeper. getApple .....	1872
Game. Gatekeeper. getAvailableHouses .....	1873
Game. Gatekeeper. getAwardRankings .....	1874
Game. Gatekeeper. getColorPalettes .....	1875

Game. Gatekeeper. getMailInBox .....	1876	Game. Nav. updateAvatar .....	1968
Game. Gatekeeper. getMailMessage .....	1877	Game. Nav. updateAvatars .....	1969
Game. Gatekeeper. getStoreItems .....	1878	Game. Nav. updateCamera .....	1970
Game. Gatekeeper. getUserLists .....	1879	Game. Nav. updateCameraDolly .....	1971
Game. Gatekeeper. goToWeb .....	1880	Game. Nav. updateCameraTruck .....	1972
Game. Gatekeeper. initUserRoom .....	1881	Game. Nav. updateFacing .....	1973
Game. Gatekeeper. inventory .....	1882	Game. Nav. WALK_SPEED .....	1960
Game. Gatekeeper. joinOK .....	1883	Game. Nav. walkTheLine .....	1974
Game. Gatekeeper. kick .....	1884	Game. NPC. Collector. fastForward .....	1921
Game. Gatekeeper. login .....	1886	Game. NPC. Collector. register .....	1922
Game. Gatekeeper. logOK .....	1885	Game. NPC. Collector. updateNPC .....	1923
Game. Gatekeeper. migrate .....	1887	Game. NPC. Cook. fastForward .....	1924
Game. Gatekeeper. newScript .....	1888	Game. NPC. Cook. register .....	1925
Game. Gatekeeper. outOfBand .....	1889	Game. NPC. Cook. updateNPC .....	1926
Game. Gatekeeper. parentApproval .....	1890	Game. NPC. CroquetPlayer. fastForward .....	1927
Game. Gatekeeper. passport .....	1891	Game. NPC. CroquetPlayer. register .....	1928
Game. Gatekeeper. playWith .....	1892	Game. NPC. CroquetPlayer. updateNPC .....	1929
Game. Gatekeeper. postman .....	1893	Game. NPC. Doodle. fastForward .....	1930
Game. Gatekeeper. prompt .....	1894	Game. NPC. Doodle. register .....	1931
Game. Gatekeeper. pub .....	1895	Game. NPC. Doodle. updateNPC .....	1932
Game. Gatekeeper. purchase .....	1896	Game. NPC. Fetcher. fastForward .....	1933
Game. Gatekeeper. quiesce .....	1897	Game. NPC. Fetcher. register .....	1934
Game. Gatekeeper. reportBug .....	1898	Game. NPC. Fetcher. updateNPC .....	1935
Game. Gatekeeper. rv .....	1899	Game. NPC. JobWorker. fastForward .....	1936
Game. Gatekeeper. scoreUpdate .....	1900	Game. NPC. JobWorker. register .....	1937
Game. Gatekeeper. sendMailMessage .....	1901	Game. NPC. JobWorker. updateNPC .....	1938
Game. Gatekeeper. serverTime .....	1902	Game. NPC. MazeBuilder. fastForward .....	1939
Game. Gatekeeper. startEvent .....	1903	Game. NPC. MazeBuilder. register .....	1940
Game. Gatekeeper. tootList .....	1904	Game. NPC. MazeBuilder. updateNPC .....	1941
Game. Gatekeeper. wardrobe .....	1905	Game. NPC. Sleeper. fastForward .....	1942
Game. Gatekeeper. wtl .....	1906	Game. NPC. Sleeper. register .....	1943
Game. GravitySystem. fastForward .....	1907	Game. NPC. Sleeper. updateNPC .....	1944
Game. GravitySystem. register .....	1908	Game. NPC. TrolleyDriver. fastForward .....	1945
Game. GravitySystem. updateEntityGravity .....	1909	Game. NPC. TrolleyDriver. register .....	1946
Game. GravitySystem. updateGravity .....	1910	Game. NPC. TrolleyDriver. updateNPC .....	1947
Game. GrowthSystem. evolve .....	1911	Game. NPC. Waiter. fastForward .....	1948
Game. GrowthSystem. fastForward .....	1912	Game. NPC. Waiter. register .....	1949
Game. GrowthSystem. grow .....	1913	Game. NPC. Waiter. updateNPC .....	1950
Game. GrowthSystem. register .....	1914	Game. NPCSystem. burgeonNPC .....	1951
Game. GrowthSystem. remove .....	1915	Game. NPCSystem. fastForward .....	1952
Game. GrowthSystem. updateGrowth .....	1916	Game. NPCSystem. initNPCs .....	1953
Game. hideWhenGameReady .....	2008	Game. NPCSystem. nextBehavior .....	1954
Game. interestingPoint .....	2009	Game. NPCSystem. register .....	1955
Game. lag .....	2010	Game. NPCSystem. updateNPC .....	1956
Game. MissileSystem. fastForward .....	1917	Game. NPCSystem. updateNPCs .....	1957
Game. MissileSystem. register .....	1918	Game. Speech. createBalloon .....	1975
Game. MissileSystem. remove .....	1919	Game. Speech. dispatchCommand .....	1976
Game. MissileSystem. updateMissiles .....	1920	Game. Speech. removeSpeech .....	1977
Game. Nav. buildWTL .....	1961	Game. Speech. say .....	1978
Game. Nav. CAMERA_MOVE_SPEED .....	1958	Game. Speech. updateSpeech .....	1979
Game. Nav. collisionP .....	1962	Game. stopSlowLoadingWatchdogs .....	2011
Game. Nav. mergeObjects .....	1963	Game. Tools. axe .....	1980
Game. Nav. moveEntityOnCourse .....	1964	Game. Tools. butterflyNet .....	1981
Game. Nav. quiesce .....	1965	Game. Tools. fishingRod .....	1982
Game. Nav. RUN_SPEED .....	1959	Game. Tools. pickaxe .....	1983
Game. Nav. runTo .....	1966	Game. Tools. sewingKit .....	1984
Game. Nav. sendWTL .....	1967	Game. Tools. shovel .....	1985
		Game. Tools. wrench .....	1986

Game. update	2012
Game. Wardrobe	1987
Game. Wardrobe. doff	1988
Game. Wardrobe. don	1989
Game. Wardrobe. drop	1990
Game. Wardrobe. finalizeExchange	1991
Game. Wardrobe. findBaseSlot	1992
Game. Wardrobe. inventory	1993
Game. Wardrobe. inventoryByKind	1994
Game. Wardrobe. proposeExchange	1995
Game. Wardrobe. readied	1996
Game. Wardrobe. readiedP	1997
Game. Wardrobe. ready	1998
Game. Wardrobe. refresh	1999
Game. Wardrobe. signExchange	2000
Game. Wardrobe. take	2001
Game. Wardrobe. wearing	2002
Game. Wardrobe. wearingP	2003
gamepadLayouts	2270
Gossip. acceptOffer	2019
Gossip. closeInfinityMode	2020
Gossip. closeStreams	2021
Gossip. connect	2022
Gossip. connectedP	2023
Gossip. createConnection	2024
Gossip. createPacket	2025
Gossip. ensureConnected	2026
Gossip. ensureKeyPair	2027
Gossip. gatekeeperAccept	2028
Gossip. getICE	2029
Gossip. getOffer	2030
Gossip. openInfinityMode	2031
Gossip. Parrot. ask	2013
Gossip. Parrot. done	2014
Gossip. Parrot. parrotErrorText	2015
Gossip. Parrot. say	2016
Gossip. Parrot. show	2017
Gossip. Parrot. ynP	2018
Gossip. send	2032
Gossip. sendLogOK	2033
Gossip. signPacket	2034
Gossip. waitForAnswer	2035
GroundBuilder. build	2036
GroundBuilder. colorForPlace	2037
GroundBuilder. initGroundPlane	2038
GroundBuilder. kinds	2039
GroundBuilder. paintPlaces	2040

**H**

host	2271
------	------

**L**

Login. acceptSignedIn	2041
Login. addChildFlag	2042
Login. addChildRequest	2043
Login. changeSensitivePlayer	2044
Login. childRequestTimeLeft	2045
Login. childSettings	2046
Login. clearTootsList	2047
Login. considerChildApproval	2048
Login. createTootListItem	2049
Login. dimUnpickedCharacters	2050
Login. disableChildMode	2051
Login. doneEditingSettings	2053
Login. doRealLogin	2052
Login. enableChildMode	2054
Login. endLoginMusic	2055
Login. fillGoogleUserInfo	2056
Login. findLIForToot	2057
Login. finishSignIn	2058
Login. firebaseLogin	2059
Login. generateNewToot	2060
Login. loadTootsList	2061
Login. loginDone	2062
Login. loginKidDirty	2063
Login. loginKidDone	2064
Login. overlay	2065
Login. pickCharacter	2066
Login. playWithCharacter	2067
Login. populateTootsList	2068
Login. quit	2069
Login. removeChildFlag	2070
Login. saveTootsList	2071
Login. serverLinkTokenToCharacter	2072
Login. setSensitiveP	2073
Login. settingsP	2074
Login. start	2075
Login. startCharacterCreation	2076
Login. startSignIn	2077
Login. storeCredentialInfo	2078
Login. switchTootsView	2079
Login. toots	2080
Login. updateNote	2081
Login. validChildCode	2082

**S**

SceneBuilder. addFurn	2083
SceneBuilder. addItem1	2084
SceneBuilder. addItem2	2085
SceneBuilder. addPlace	2086
SceneBuilder. addText	2087
SceneBuilder. build	2088
SkyBuilder. build	2089
SkyBuilder. buildMatchingSky	2090
SkyBuilder. buildMatchingWeather	2091
SkyBuilder. setCloudCover	2092
SkyBuilder. setFirstSkyLayer	2093
SkyBuilder. setMoon	2094

SkyBuilder. setPlanet .....	2095
SkyBuilder. setPrecipitation .....	2096
SkyBuilder. setStarfield .....	2097
SkyBuilder. setSun .....	2098
SkyBuilder. setTheMoon .....	2099
SkyBuilder. setTheOtherMoon .....	2100
SkyBuilder. setThePinkMoon .....	2101
SkyBuilder. sunX .....	2102
SkyBuilder. sunY .....	2103
SkyBuilder. update .....	2104
SkyBuilder. updateSkyData .....	2105

## T

Tank. afterRender .....	2109
Tank. attachmentOverlaysNeedUpdateP .....	2110
Tank. CameraManager. positionCameraForAvatarCloseUp .....	2106
Tank. CameraManager. positionCameraForAvatarViewer .....	2107
Tank. CameraManager. positionCameraForGameBoard .....	2108
Tank. createScene .....	2111
Tank. destroyAvatar .....	2112
Tank. findAvatar .....	2113
Tank. getCanvas .....	2114
Tank. getLargestChildMesh .....	2115
Tank. init3DEngine .....	2116
Tank. initArcCamera .....	2117
Tank. initOTSCamera .....	2118
Tank. initPlayerToot .....	2119
Tank. initScene .....	2120
Tank. loadUISounds .....	2121
Tank. playerAvatar .....	2122
Tank. prepareFor3D .....	2123
Tank. shutDown .....	2124
Tank. start3D .....	2125
Tank. start3DReal .....	2126
Tank. startRenderLoop .....	2127
Tank. updateAvatarFor .....	2128
Tank. updateCamera .....	2129

## U

UI. Audio. context .....	2130
UI. Audio. gainNode .....	2131
UI. Audio. setVolume .....	2132
UI. Audio. updateVolumeMuteIcon .....	2133
UI. Audio. updateVolumeSlider .....	2134
UI. Audio. updateVolumeUI .....	2135
UI. Audio. volumeDown .....	2136
UI. Audio. volumeMute .....	2137
UI. Audio. volumeUp .....	2138
UI. clickedOnItem .....	2241
UI. commands .....	2242
UI. confirmPretty .....	2243
UI. findAdjacentEntity .....	2244
UI. forceQuit .....	2245

UI. Gamepad. addGamepad .....	2140
UI. Gamepad. axisUpdate .....	2141
UI. Gamepad. buttonEvent .....	2142
UI. Gamepad. connectHandler .....	2143
UI. Gamepad. controllers .....	2145
UI. Gamepad. controllerState .....	2144
UI. Gamepad. disconnectHandler .....	2146
UI. Gamepad. removeGamepad .....	2147
UI. Gamepad. ROTATION_SPEED .....	2139
UI. Gamepad. scanGamepads .....	2148
UI. Gamepad. updateStatus .....	2149
UI. HUD. beginWatchingPaperdollWindowForClose ..	2150
UI. HUD. clickedOnMesh .....	2151
UI. HUD. closePanel .....	2152
UI. HUD. closeTalkBox .....	2153
UI. HUD. connectTalkBox .....	2154
UI. HUD. convertCanvasEventTo3D .....	2155
UI. HUD. createHUDLoaderPanel .....	2156
UI. HUD. createPaperdollCanvas .....	2157
UI. HUD. destroyHUD .....	2158
UI. HUD. dropHUDPanels .....	2159
UI. HUD. getOpenPanel .....	2160
UI. HUD. initHUD .....	2161
UI. HUD. loadHTML .....	2162
UI. HUD. loadHUDPanel .....	2163
UI. HUD. loadScriptIntoDiv .....	2164
UI. HUD. nameTagClicked .....	2165
UI. HUD. openPaperdoll .....	2166
UI. HUD. openTalkBox .....	2167
UI. HUD. paperdollCurrentP .....	2168
UI. HUD. positionPaperdollMini .....	2169
UI. HUD. refreshAttachmentOverlays .....	2170
UI. HUD. refreshAttachmentsForAvatar .....	2171
UI. HUD. refreshEquipment .....	2172
UI. HUD. refreshHUD .....	2173
UI. HUD. refreshNameTagAttachment .....	2174
UI. HUD. refreshPaperdoll .....	2175
UI. HUD. refreshSpeechAttachment .....	2176
UI. HUD. refreshTalkStatus .....	2177
UI. HUD. refreshTimeLeft .....	2178
UI. HUD. refreshWallet .....	2179
UI. HUD. returnPaperdollMini .....	2180
UI. HUD. setPaperdollForPlayerAvatar .....	2181
UI. HUD. showCamera .....	2182
UI. HUD. showControlPanel .....	2183
UI. HUD. showHUDPanel .....	2184
UI. HUD. showMobile .....	2185
UI. HUD. showPlayerCard .....	2186
UI. HUD. switchActiveItem .....	2187
UI. HUD. talkBoxOpenP .....	2188
UI. HUD. toggleElement .....	2189
UI. HUD. toggleHUDPanel .....	2190
UI. HUD. toggleTalkBox .....	2191
UI. HUD. toggleTalkEmoji .....	2192
UI. HUD. toggleTalkExpression .....	2193
UI. HUD. toggleTalkLoud .....	2194
UI. insertEmoji .....	2246





“It’s always dark at the beginning . . . ”