



# **Red Hat Satellite 5.7**

## **API Guide**

A reference guide to the Red Hat Satellite API



# Red Hat Satellite 5.7 API Guide

---

A reference guide to the Red Hat Satellite API

Red Hat Satellite Documentation Team

## Legal Notice

Copyright © 2014 Red Hat.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This book provides an introduction to the XML-RPC API for Red Hat Satellite, and includes several examples of its use.

## Table of Contents

<b>CHAPTER 1. INTRODUCTION TO RED HAT SATELLITE .....</b>	<b>40</b>
1.1. THE RED HAT SATELLITE API .....	40
1.2. USING XML-RPC WITH THE RED HAT SATELLITE API .....	40
1.3. USING THE READ-ONLY API .....	41
1.3.1. Creating a Read-only User .....	42
<b>CHAPTER 2. EXAMPLES .....</b>	<b>43</b>
2.1. PYTHON EXAMPLES .....	43
2.2. PERL EXAMPLE .....	44
2.3. RUBY EXAMPLE .....	44
<b>PART I. METHODS .....</b>	<b>46</b>
<b>CHAPTER 3. NAMESPACE: ACTIONCHAIN .....</b>	<b>47</b>
3.1. METHOD: ADDCONFIGURATIONDEPLOYMENT .....	47
Parameters .....	47
Returns .....	47
3.2. METHOD: ADDPACKAGEINSTALL .....	47
Parameters .....	47
Returns .....	47
3.3. METHOD: ADDPACKAGEREMOVAL .....	47
Parameters .....	47
Returns .....	48
3.4. METHOD: ADDPACKAGEUPGRADE .....	48
Parameters .....	48
Returns .....	48
3.5. METHOD: ADDPACKAGEVERIFY .....	48
Parameters .....	48
Returns .....	48
3.6. METHOD: ADDSCRIPTRUN .....	48
Parameters .....	48
Returns .....	49
3.7. METHOD: ADDSYSTEMREBOOT .....	49
Parameters .....	49
Returns .....	49
3.8. METHOD: CREATECHAIN .....	49
Parameters .....	49
Returns .....	49
3.9. METHOD: DELETECHAIN .....	49
Parameters .....	49
Returns .....	50
3.10. METHOD: LISTCHAINACTIONS .....	50
Parameters .....	50
Returns .....	50
3.11. METHOD: LISTCHAINS .....	50
Parameters .....	50
Returns .....	50
3.12. METHOD: REMOVEACTION .....	50
Parameters .....	51
Returns .....	51
3.13. METHOD: RENAMECHAIN .....	51
Parameters .....	51

Returns	51
3.14. METHOD: SCHEDULECHAIN	51
Parameters	51
Returns	51
<b>CHAPTER 4. NAMESPACE: ACTIVATIONKEY .....</b>	<b>52</b>
4.1. METHOD: ADDCHILDCHANNELS	52
Parameters	52
Returns	52
4.2. METHOD: ADDCONFIGCHANNELS	52
Parameters	52
Returns	52
4.3. METHOD: ADDENTITLEMENTS	52
Parameters	53
Returns	53
4.4. METHOD: ADDPACKAGENAMES	53
Parameters	53
Returns	53
4.5. METHOD: ADDPACKAGES	53
Parameters	53
Returns	54
4.6. METHOD: ADDSERVERGROUPS	54
Parameters	54
Returns	54
4.7. METHOD: CHECKCONFIGDEPLOYMENT	54
Parameters	54
Returns	54
4.8. METHOD: CLONE	54
Parameters	54
Returns	55
4.9. METHOD: CREATE	55
Parameters	55
Returns	55
4.10. METHOD: CREATE	55
Parameters	55
Returns	56
4.11. METHOD: DELETE	56
Parameters	56
Returns	56
4.12. METHOD: DISABLECONFIGDEPLOYMENT	56
Parameters	56
Returns	56
4.13. METHOD: ENABLECONFIGDEPLOYMENT	56
Parameters	56
Returns	57
4.14. METHOD: GETDETAILS	57
Parameters	57
Returns	57
4.15. METHOD: LISTACTIVATEDSYSTEMS	58
Parameters	58
Returns	58
4.16. METHOD: LISTACTIVATIONKEYS	58
Parameters	58

Returns	58
4.17. METHOD: LISTCONFIGCHANNELS	59
Parameters	59
Returns	59
4.18. METHOD: REMOVECHILDCHANNELS	60
Parameters	60
Returns	60
4.19. METHOD: REMOVECONFIGCHANNELS	60
Parameters	60
Returns	60
4.20. METHOD: REMOVEENTITLEMENTS	60
Parameters	60
Returns	61
4.21. METHOD: REMOVEPACKAGENAMES	61
Parameters	61
Returns	61
4.22. METHOD: REMOVEPACKAGES	61
Parameters	61
Returns	61
4.23. METHOD: REMOVESERVERGROUPS	62
Parameters	62
Returns	62
4.24. METHOD: SETCONFIGCHANNELS	62
Parameters	62
Returns	62
4.25. METHOD: SETDETAILS	62
Parameters	62
Returns	63
<b>CHAPTER 5. NAMESPACE: API .....</b>	<b>64</b>
5.1. METHOD: GETAPICALLLIST	64
Parameters	64
Returns	64
5.2. METHOD: GETAPINAMESPACECALLLIST	64
Parameters	64
Returns	64
5.3. METHOD: GETAPINAMESPACES	64
Parameters	64
Returns	65
5.4. METHOD: GETVERSION	65
Parameters	65
Returns	65
5.5. METHOD: SYSTEMVERSION	65
Parameters	65
Returns	65
<b>CHAPTER 6. NAMESPACE: AUTH .....</b>	<b>66</b>
6.1. METHOD: LOGIN	66
Parameters	66
Returns	66
6.2. METHOD: LOGIN	66
Parameters	66
Returns	66

6.3. METHOD: LOGOUT	66
Parameters	66
Returns	66
<b>CHAPTER 7. NAMESPACE: CHANNEL .....</b>	<b>67</b>
7.1. METHOD: LISTALLCHANNELS	67
Parameters	67
Returns	67
7.2. METHOD: LISTMYCHANNELS	67
Parameters	67
Returns	67
7.3. METHOD: LISTPOPULARCHANNELS	68
Parameters	68
Returns	68
7.4. METHOD: LISTREDHATCHANNELS	68
Parameters	68
Returns	68
7.5. METHOD: LISTRETIREDCHANNELS	69
Parameters	69
Returns	69
7.6. METHOD: LISTSHAREDCHANNELS	69
Parameters	69
Returns	69
7.7. METHOD: LISTSOFTWARECHANNELS	70
Parameters	70
Returns	70
7.8. METHOD: LISTVENDORCHANNELS	70
Parameters	70
Returns	70
<b>CHAPTER 8. NAMESPACE: CHANNEL.ACCESS .....</b>	<b>72</b>
8.1. METHOD: DISABLEUSERRESTRICTIONS	72
Parameters	72
Returns	72
8.2. METHOD: ENABLEUSERRESTRICTIONS	72
Parameters	72
Returns	72
8.3. METHOD: GETORGSHARING	72
Parameters	72
Returns	72
8.4. METHOD: SETORGSHARING	72
Parameters	72
Returns	73
<b>CHAPTER 9. NAMESPACE: CHANNEL.ORG .....</b>	<b>74</b>
9.1. METHOD: DISABLEACCESS	74
Parameters	74
Returns	74
9.2. METHOD: ENABLEACCESS	74
Parameters	74
Returns	74
9.3. METHOD: LIST	74
Parameters	74
Returns	74



<b>CHAPTER 10. NAMESPACE: CHANNELSOFTWARE .....</b>	<b>76</b>
10.1. METHOD: ADDPACKAGES	76
Parameters	76
Returns	76
10.2. METHOD: ADDREPOFILTER	76
Parameters	76
Returns	76
10.3. METHOD: ASSOCIATEREPO	76
Parameters	76
Returns	76
10.4. METHOD: AVAILABLEENTITLEMENTS	77
Parameters	78
Returns	78
10.5. METHOD: CLEARREPOFILTERS	78
Parameters	78
Returns	78
10.6. METHOD: CLONE	78
Parameters	78
Returns	79
10.7. METHOD: CREATE	79
Parameters	79
Returns	79
10.8. METHOD: CREATE	79
Parameters	79
Returns	80
10.9. METHOD: CREATE	80
Parameters	80
Returns	80
10.10. METHOD: CREATEREPO	80
Parameters	80
Returns	81
10.11. METHOD: DELETE	81
Parameters	81
Returns	81
10.12. METHOD: DISASSOCIATEREPO	81
Parameters	81
Returns	81
10.13. METHOD: GETCHANNELLASTBUILDID	82
Parameters	82
Returns	82
10.14. METHOD: GETDETAILS	83
Parameters	83
Returns	83
10.15. METHOD: GETDETAILS	84
Parameters	84
Returns	84
10.16. METHOD: GETREPODETAILS	85
Parameters	85
Returns	85
10.17. METHOD: GETREPODETAILS	85
Parameters	85
Returns	85
10.18. METHOD: GETREPOSYNCCRONEXPRESSION	86

Parameters	86
Returns	86
10.19. METHOD: ISGLOBALLYSUBSCRIBABLE	86
Parameters	86
Returns	86
10.20. METHOD: ISUSERMANAGEABLE	86
Parameters	86
Returns	86
10.21. METHOD: ISUSERSUBSCRIBABLE	86
Parameters	87
Returns	87
10.22. METHOD: LISTALLPACKAGES	87
Parameters	87
Returns	87
10.23. METHOD: LISTALLPACKAGES	87
Parameters	88
Returns	88
10.24. METHOD: LISTALLPACKAGES	88
Parameters	88
Returns	88
10.25. METHOD: LISTALLPACKAGES	89
Parameters	89
Returns	89
10.26. METHOD: LISTALLPACKAGES	90
Parameters	90
Returns	90
10.27. METHOD: LISTALLPACKAGESBYDATE	90
Parameters	90
Returns	91
10.28. METHOD: LISTALLPACKAGESBYDATE	91
Parameters	91
Returns	91
10.29. METHOD: LISTALLPACKAGESBYDATE	92
Parameters	92
Returns	92
10.30. METHOD: LISTARCHES	92
Parameters	92
Returns	92
10.31. METHOD: LISTCHANNELREPOS	93
Parameters	93
Returns	93
10.32. METHOD: LISTCHILDREN	93
Parameters	93
Returns	93
10.33. METHOD: LISTERRATA	94
Parameters	94
Returns	94
10.34. METHOD: LISTERRATA	95
Parameters	95
Returns	95
10.35. METHOD: LISTERRATA	95
Parameters	95
Returns	95

10.36. METHOD: LISTERRATA	96
Parameters	96
Returns	96
10.37. METHOD: LISTERRATA	97
Parameters	97
Returns	97
10.38. METHOD: LISTERRATABYTYPE	97
Parameters	97
Returns	97
10.39. METHOD: LISTERRATANEEDINGSYNC	98
Parameters	98
Returns	98
10.40. METHOD: LISTLATESTPACKAGES	98
Parameters	98
Returns	98
10.41. METHOD: LISTPACKAGESWITHOUTCHANNEL	99
Parameters	99
Returns	99
10.42. METHOD: LISTREPOFILTERS	99
Parameters	99
Returns	100
10.43. METHOD: LISTSUBSCRIBEDSYSTEMS	100
Parameters	100
Returns	100
10.44. METHOD: LISTSYSTEMCHANNELS	100
Parameters	100
Returns	100
10.45. METHOD: LISTUSERREPOS	101
Parameters	101
Returns	101
10.46. METHOD: MERGEERRATA	101
Parameters	101
Returns	101
10.47. METHOD: MERGEERRATA	101
Parameters	101
Returns	102
10.48. METHOD: MERGEERRATA	102
Parameters	102
Returns	102
10.49. METHOD: MERGEPACKAGES	103
Parameters	103
Returns	103
10.50. METHOD: REGENERATENEEEDEDCACHE	103
Parameters	103
Returns	103
10.51. METHOD: REGENERATENEEEDEDCACHE	104
Parameters	104
Returns	104
10.52. METHOD: REGENERATEYUMCACHE	104
Parameters	104
Returns	104
10.53. METHOD: REMOVEERRATA	104
Parameters	104

Returns	104
10.54. METHOD: REMOVEPACKAGES	104
Parameters	104
Returns	105
10.55. METHOD: REMOVEREPO	105
Parameters	105
Returns	105
10.56. METHOD: REMOVEREPO	105
Parameters	105
Returns	105
10.57. METHOD: REMOVEREPOFILTER	105
Parameters	105
Returns	105
10.58. METHOD: SETCONTACTDETAILS	106
Parameters	106
Returns	106
10.59. METHOD: SETDETAILS	106
Parameters	106
Returns	107
10.60. METHOD: SETGLOBALLYSUBSCRIBABLE	107
Parameters	107
Returns	107
10.61. METHOD: SETREPOFILTERS	107
Parameters	107
Returns	107
10.62. METHOD: SETSYSTEMCHANNELS	107
Parameters	107
Returns	108
10.63. METHOD: SETUSERMANAGEABLE	108
Parameters	108
Returns	108
10.64. METHOD: SETUSERSUBSCRIBABLE	108
Parameters	108
Returns	108
10.65. METHOD: SUBSCRIBESYSTEM	108
Parameters	109
Returns	109
10.66. METHOD: SYNCERRATA	109
Parameters	109
Returns	109
10.67. METHOD: SYNCREPO	109
Parameters	109
Returns	109
10.68. METHOD: SYNCREPO	109
Parameters	109
Returns	110
10.69. METHOD: SYNCREPO	110
Parameters	110
Returns	110
10.70. METHOD: SYNCREPO	110
Parameters	110
Returns	110
10.71. METHOD: UPDATEREPO	110

Parameters	111
Returns	111
10.72. METHOD: UPDATEREPOLABEL	111
Parameters	111
Returns	111
10.73. METHOD: UPDATEREPOLABEL	111
Parameters	111
Returns	112
10.74. METHOD: UPDATEREPOURL	112
Parameters	112
Returns	112
10.75. METHOD: UPDATEREPOURL	112
Parameters	112
Returns	112
<b>CHAPTER 11. NAMESPACE: CONFIGCHANNEL .....</b>	<b>114</b>
11.1. METHOD: CHANNELEXISTS	114
Parameters	114
Returns	114
11.2. METHOD: CREATE	114
Parameters	114
Returns	114
11.3. METHOD: CREATEORUPDATEPATH	115
Parameters	115
Returns	115
11.4. METHOD: CREATEORUPDATESYMLINK	116
Parameters	116
Returns	116
11.5. METHOD: DELETECHANNELS	117
Parameters	117
Returns	118
11.6. METHOD: DELETEDFILEREVISIONS	118
Parameters	118
Returns	118
11.7. METHOD: DELETEDFILES	118
Parameters	118
Returns	118
11.8. METHOD: DEPLOYALLSYSTEMS	118
Parameters	118
Returns	119
11.9. METHOD: DEPLOYALLSYSTEMS	119
Parameters	119
Returns	119
11.10. METHOD: DEPLOYALLSYSTEMS	119
Parameters	119
Returns	119
11.11. METHOD: DEPLOYALLSYSTEMS	119
Parameters	119
Returns	119
11.12. METHOD: GETDETAILS	120
Parameters	120
Returns	120
11.13. METHOD: GETDETAILS	120

Parameters	120
Returns	120
11.14. METHOD: GETENCODEDFILEREVISION	121
Parameters	121
Returns	121
11.15. METHOD: GETFILEREVISION	122
Parameters	122
Returns	122
11.16. METHOD: GETFILEREVISIONS	123
Parameters	123
Returns	123
11.17. METHOD: LISTFILES	124
Parameters	124
Returns	124
11.18. METHOD: LISTGLOBALS	124
Parameters	124
Returns	125
11.19. METHOD: LISTSUBSCRIBEDSYSTEMS	125
Parameters	125
Returns	125
11.20. METHOD: LOOKUPCHANNELINFO	125
Parameters	125
Returns	126
11.21. METHOD: LOOKUPFILEINFO	126
Parameters	126
Returns	126
11.22. METHOD: LOOKUPFILEINFO	127
Parameters	127
Returns	127
11.23. METHOD: SCHEDULEFILECOMPARISONS	128
Parameters	128
Returns	129
11.24. METHOD: UPDATE	129
Parameters	129
Returns	129
<b>CHAPTER 12. NAMESPACE: DISTCHANNEL .....</b>	<b>130</b>
12.1. METHOD: LISTDEFAULTMAPS	130
Parameters	130
Returns	130
12.2. METHOD: LISTMAPSFORORG	130
Parameters	130
Returns	130
12.3. METHOD: LISTMAPSFORORG	130
Parameters	131
Returns	131
12.4. METHOD: SETMAPFORORG	131
Parameters	131
Returns	131
<b>CHAPTER 13. NAMESPACE: ERRATA .....</b>	<b>132</b>
13.1. METHOD: ADDPACKAGES	132
Parameters	132

Returns	132
13.2. METHOD: APPLICABLETOCHANNELS	132
Parameters	132
Returns	132
13.3. METHOD: BUGZILLAFIXES	132
Parameters	132
Returns	133
13.4. METHOD: CLONE	133
Parameters	133
Returns	133
13.5. METHOD: CLONEASORIGINAL	133
Parameters	133
Returns	134
13.6. METHOD: CLONEASORIGINALASYNC	134
Parameters	134
Returns	134
13.7. METHOD: CLONEASYNC	134
Parameters	134
Returns	134
13.8. METHOD: CREATE	134
Parameters	135
Returns	135
13.9. METHOD: DELETE	136
Parameters	136
Returns	136
13.10. METHOD: FINDBYCVE	136
Parameters	136
Returns	136
13.11. METHOD: GETDETAILS	136
Parameters	137
Returns	137
13.12. METHOD: LISTAFFECTEDSYSTEMS	137
Parameters	137
Returns	137
13.13. METHOD: LISTBYDATE	138
Parameters	138
Returns	138
13.14. METHOD: LISTCVES	138
Parameters	138
Returns	138
13.15. METHOD: LISTKEYWORDS	138
Parameters	138
Returns	139
13.16. METHOD: LISTPACKAGES	139
Parameters	139
Returns	139
13.17. METHOD: LISTUNPUBLISHEDERRATA	140
Parameters	140
Returns	140
13.18. METHOD: PUBLISH	140
Parameters	140
Returns	141
13.19. METHOD: PUBLISHASORIGINAL	141

Parameters	141
Returns	141
13.20. METHOD: REMOVEPACKAGES	141
Parameters	141
Returns	142
13.21. METHOD: SETDETAILS	142
Parameters	142
Returns	143
<b>CHAPTER 14. NAMESPACE: KICKSTART .....</b>	<b>144</b>
14.1. METHOD: CLONEPROFILE	144
Parameters	144
Returns	144
14.2. METHOD: CREATEPROFILE	144
Parameters	144
Returns	144
14.3. METHOD: CREATEPROFILE	144
Parameters	144
Returns	145
14.4. METHOD: CREATEPROFILEWITHCUSTOMURL	145
Parameters	145
Returns	145
14.5. METHOD: CREATEPROFILEWITHCUSTOMURL	145
Parameters	145
Returns	146
14.6. METHOD: DELETEPROFILE	146
Parameters	146
Returns	146
14.7. METHOD: DISABLEPROFILE	146
Parameters	146
Returns	146
14.8. METHOD: FINDKICKSTARTFORIP	146
Parameters	146
Returns	146
14.9. METHOD: IMPORTFILE	147
Parameters	147
Returns	147
14.10. METHOD: IMPORTFILE	147
Parameters	147
Returns	147
14.11. METHOD: IMPORTFILE	147
Parameters	147
Returns	148
14.12. METHOD: IMPORTRAWFILE	148
Parameters	148
Returns	148
14.13. METHOD: IMPORTRAWFILE	148
Parameters	148
Returns	149
14.14. METHOD: ISPROFILEDISABLED	149
Parameters	149
Returns	149
14.15. METHOD: LISTALLIPRANGES	149



Parameters	149
Returns	149
14.16. METHOD: LISTKICKSTARTABLECHANNELS	149
Parameters	149
Returns	149
14.17. METHOD: LISTKICKSTARTABLETREES	150
Parameters	151
Returns	151
14.18. METHOD: LISTKICKSTARTS	151
Parameters	151
Returns	151
14.19. METHOD: RENAMEPROFILE	151
Parameters	151
Returns	152
<b>CHAPTER 15. NAMESPACE: KICKSTART.FILEPRESERVATION</b>	<b>153</b>
15.1. METHOD: CREATE	153
Parameters	153
Returns	153
15.2. METHOD: DELETE	153
Parameters	153
Returns	153
15.3. METHOD: GETDETAILS	153
Parameters	153
Returns	153
15.4. METHOD: LISTALLFILEPRESERVATIONS	154
Parameters	154
Returns	154
<b>CHAPTER 16. NAMESPACE: KICKSTART.KEYS</b>	<b>155</b>
16.1. METHOD: CREATE	155
Parameters	155
Returns	155
16.2. METHOD: DELETE	155
Parameters	155
Returns	155
16.3. METHOD: GETDETAILS	155
Parameters	155
Returns	155
16.4. METHOD: LISTALLKEYS	156
Parameters	156
Returns	156
16.5. METHOD: UPDATE	156
Parameters	156
Returns	156
<b>CHAPTER 17. NAMESPACE: KICKSTART.PROFILE.KEYS</b>	<b>157</b>
17.1. METHOD: ADDACTIVATIONKEY	157
Parameters	157
Returns	157
17.2. METHOD: GETACTIVATIONKEYS	157
Parameters	157
Returns	157
17.3. METHOD: REMOVEACTIVATIONKEY	158

Parameters	158
Returns	158
<b>CHAPTER 18. NAMESPACE: KICKSTART.PROFILE.SOFTWARE .....</b>	<b>159</b>
18.1. METHOD: APPENDTOSOFTWARELIST	159
Parameters	159
Returns	159
18.2. METHOD: GETSOFTWAREDETAILS	159
Parameters	159
Returns	159
18.3. METHOD: GETSOFTWARELIST	159
Parameters	159
Returns	159
18.4. METHOD: SETSOFTWAREDETAILS	159
Parameters	160
Returns	160
18.5. METHOD: SETSOFTWARELIST	160
Parameters	160
Returns	160
18.6. METHOD: SETSOFTWARELIST	160
Parameters	160
Returns	160
<b>CHAPTER 19. NAMESPACE: KICKSTART.PROFILE.SYSTEM .....</b>	<b>161</b>
19.1. METHOD: ADDFILEPRESERVATIONS	161
Parameters	161
Returns	161
19.2. METHOD: ADDKEYS	161
Parameters	161
Returns	161
19.3. METHOD: CHECKCONFIGMANAGEMENT	161
Parameters	161
Returns	161
19.4. METHOD: CHECKREMOTECOMMANDS	162
Parameters	162
Returns	162
19.5. METHOD: DISABLECONFIGMANAGEMENT	162
Parameters	162
Returns	162
19.6. METHOD: DISABLEREMOTECOMMANDS	162
Parameters	162
Returns	162
19.7. METHOD: ENABLECONFIGMANAGEMENT	162
Parameters	162
Returns	163
19.8. METHOD: ENABLEREMOTECOMMANDS	163
Parameters	163
Returns	163
19.9. METHOD: GETLOCALE	163
Parameters	163
Returns	163
19.10. METHOD: GETPARTITIONINGScheme	163
Parameters	163

Returns	163
19.11. METHOD: GETREGISTRATIONTYPE	164
Parameters	164
Returns	164
19.12. METHOD: GETSELINUX	164
Parameters	164
Returns	164
19.13. METHOD: LISTFILEPRESERVATIONS	164
Parameters	164
Returns	164
19.14. METHOD: LISTKEYS	165
Parameters	165
Returns	165
19.15. METHOD: REMOVEFILEPRESERVATIONS	165
Parameters	165
Returns	165
19.16. METHOD: REMOVEKEYS	165
Parameters	165
Returns	166
19.17. METHOD: SETLOCALE	166
Parameters	166
Returns	166
19.18. METHOD: SETPARTITIONINGScheme	166
Parameters	166
Returns	166
19.19. METHOD: SETREGISTRATIONTYPE	166
Parameters	167
Returns	167
19.20. METHOD: SETSELINUX	167
Parameters	167
Returns	167
<b>CHAPTER 20. NAMESPACE: KICKSTART.PROFILE .....</b>	<b>168</b>
20.1. METHOD: ADDIPRANGE	168
Parameters	168
Returns	168
20.2. METHOD: ADDSCRIPT	168
Parameters	168
Returns	168
20.3. METHOD: ADDSCRIPT	168
Parameters	168
Returns	169
20.4. METHOD: ADDSCRIPT	169
Parameters	169
Returns	169
20.5. METHOD: COMPAREACTIVATIONKEYS	169
Parameters	170
Returns	170
20.6. METHOD: COMPAREADVANCEDOPTIONS	171
Parameters	171
Returns	171
20.7. METHOD: COMPAREPACKAGES	172
Parameters	172

Returns	172
20.8. METHOD: DOWNLOADKICKSTART	172
Parameters	172
Returns	173
20.9. METHOD: DOWNLOADRENDEREDKICKSTART	173
Parameters	173
Returns	173
20.10. METHOD: GETADVANCEDOPTIONS	173
Parameters	173
Returns	173
20.11. METHOD: GETAVAILABLEREPOSITORIES	173
Parameters	173
Returns	174
20.12. METHOD: GETCFGPRESERVATION	174
Parameters	174
Returns	174
20.13. METHOD: GETCHILDCHANNELS	174
Parameters	174
Returns	174
20.14. METHOD: GETCUSTOMOPTIONS	174
Parameters	174
Returns	174
20.15. METHOD: GETKICKSTARTTREE	175
Parameters	175
Returns	175
20.16. METHOD: GETREPOSITORIES	175
Parameters	175
Returns	175
20.17. METHOD: GETUPDATETYPE	175
Parameters	175
Returns	175
20.18. METHOD: GETVARIABLES	175
Parameters	175
Returns	176
20.19. METHOD: GETVIRTUALIZATIONTYPE	176
Parameters	176
Returns	176
20.20. METHOD: LISTIPRANGES	176
Parameters	176
Returns	176
20.21. METHOD: LISTSCRIPTS	176
Parameters	176
Returns	177
20.22. METHOD: ORDERSSCRIPTS	177
Parameters	177
Returns	177
20.23. METHOD: REMOVEIPRANGE	177
Parameters	178
Returns	178
20.24. METHOD: REMOVESCRIPT	178
Parameters	178
Returns	178
20.25. METHOD: SETADVANCEDOPTIONS	178

Parameters	178
Returns	179
20.26. METHOD: SETCFGPRESERVATION	179
Parameters	179
Returns	179
20.27. METHOD: SETCHILDCHANNELS	179
Parameters	179
Returns	179
20.28. METHOD: SETCUSTOMOPTIONS	179
Parameters	179
Returns	179
20.29. METHOD: SETKICKSTARTTREE	179
Parameters	180
Returns	180
20.30. METHOD: SETLOGGING	180
Parameters	180
Returns	180
20.31. METHOD: SETREPOSITORIES	180
Parameters	180
Returns	180
20.32. METHOD: SETUPDATETYPE	180
Parameters	180
Returns	181
20.33. METHOD: SETVARIABLES	181
Parameters	181
Returns	181
20.34. METHOD: SETVIRTUALIZATIONTYPE	181
Parameters	181
Returns	181
<b>CHAPTER 21. NAMESPACE: KICKSTART.SNIPPET .....</b>	<b>182</b>
21.1. METHOD: CREATEORUPDATE	182
Parameters	182
Returns	182
21.2. METHOD: DELETE	182
Parameters	182
Returns	182
21.3. METHOD: LISTALL	182
Parameters	182
Returns	182
21.4. METHOD: LISTCUSTOM	183
Parameters	183
Returns	183
21.5. METHOD: LISTDEFAULT	183
Parameters	183
Returns	183
<b>CHAPTER 22. NAMESPACE: KICKSTART.TREE .....</b>	<b>185</b>
22.1. METHOD: CREATE	185
Parameters	185
Returns	185
22.2. METHOD: DELETE	185
Parameters	185

Returns	185
22.3. METHOD: DELETETREEANDPROFILES	185
Parameters	185
Returns	185
22.4. METHOD: GETDETAILS	186
Parameters	186
Returns	186
22.5. METHOD: LIST	186
Parameters	186
Returns	186
22.6. METHOD: LISTINSTALLTYPES	186
Parameters	187
Returns	187
22.7. METHOD: RENAME	187
Parameters	187
Returns	187
22.8. METHOD: UPDATE	187
Parameters	187
Returns	187
<b>CHAPTER 23. NAMESPACE: ORG .....</b>	<b>188</b>
23.1. METHOD: CREATE	188
Parameters	188
Returns	188
23.2. METHOD: DELETE	189
Parameters	189
Returns	189
23.3. METHOD: GETCRASHFILESIZELIMIT	189
Parameters	189
Returns	189
23.4. METHOD: GETDETAILS	189
Parameters	189
Returns	189
23.5. METHOD: GETDETAILS	190
Parameters	190
Returns	190
23.6. METHOD: GETPOLICYFORSCAPFILEUPLOAD	190
Parameters	190
Returns	191
23.7. METHOD: GETPOLICYFORSCAPRESULTDELETION	191
Parameters	191
Returns	191
23.8. METHOD: ISCRASHREPORTINGENABLED	191
Parameters	191
Returns	191
23.9. METHOD: ISCRASHFILEUPLOADENABLED	191
Parameters	191
Returns	191
23.10. METHOD: LISTORGS	192
Parameters	192
Returns	192
23.11. METHOD: LISTSOFTWAREENTITLEMENTS	192
Parameters	192

Returns	192
23.12. METHOD: LISTSOFTWAREENTITLEMENTS	193
Parameters	193
Returns	193
23.13. METHOD: LISTSOFTWAREENTITLEMENTS	193
Parameters	193
Returns	194
23.14. METHOD: LISTSOFTWAREENTITLEMENTSFORORG	194
Parameters	194
Returns	194
23.15. METHOD: LISTSYSTEMENTITLEMENTS	195
Parameters	195
Returns	195
23.16. METHOD: LISTSYSTEMENTITLEMENTS	195
Parameters	195
Returns	195
23.17. METHOD: LISTSYSTEMENTITLEMENTS	196
Parameters	196
Returns	196
23.18. METHOD: LISTSYSTEMENTITLEMENTSFORORG	196
Parameters	196
Returns	196
23.19. METHOD: LISTUSERS	197
Parameters	197
Returns	197
23.20. METHOD: MIGRATESYSTEMS	197
Parameters	197
Returns	198
23.21. METHOD: SETCRASHFILESIZELIMIT	198
Parameters	198
Returns	198
23.22. METHOD: SETCRASHREPORTING	198
Parameters	198
Returns	198
23.23. METHOD: SETCRASHFILEUPLOAD	198
Parameters	198
Returns	199
23.24. METHOD: SETPOLICYFORSCAPFILEUPLOAD	199
Parameters	199
Returns	199
23.25. METHOD: SETPOLICYFORSCAPRESULTDELETION	199
Parameters	199
Returns	199
23.26. METHOD: SETSOFTWAREENTITLEMENTS	199
Parameters	199
Returns	200
23.27. METHOD: SETSOFTWAREFLEXENTITLEMENTS	200
Parameters	200
Returns	200
23.28. METHOD: SETSYSTEMENTITLEMENTS	200
Parameters	200
Returns	200
23.29. METHOD: UPDATENAME	201

Parameters	201
Returns	201
<b>CHAPTER 24. NAMESPACE: ORG.TRUSTS .....</b>	<b>202</b>
24.1. METHOD: ADDTRUST	202
Parameters	202
Returns	202
24.2. METHOD: GETDETAILS	202
Parameters	202
Returns	202
24.3. METHOD: LISTCHANNELSCONSUMED	202
Parameters	202
Returns	203
24.4. METHOD: LISTCHANNELSPROVIDED	203
Parameters	203
Returns	203
24.5. METHOD: LISTORGS	203
Parameters	203
Returns	203
24.6. METHOD: LISTSYSTEMSAFFECTED	204
Parameters	204
Returns	204
24.7. METHOD: LISTTRUSTS	204
Parameters	204
Returns	204
24.8. METHOD: REMOVETRUST	204
Parameters	204
Returns	205
<b>CHAPTER 25. NAMESPACE: PACKAGES .....</b>	<b>206</b>
25.1. METHOD: FINDBYNVREA	206
Parameters	206
Returns	206
25.2. METHOD: GETDETAILS	206
Parameters	206
Returns	207
25.3. METHOD: GETPACKAGE	207
Parameters	208
Returns	208
25.4. METHOD: GETPACKAGEURL	208
Parameters	208
Returns	208
25.5. METHOD: LISTCHANGELOG	208
Parameters	208
Returns	208
25.6. METHOD: LISTDEPENDENCIES	208
Parameters	208
Returns	208
25.7. METHOD: LISTFILES	209
Parameters	209
Returns	209
25.8. METHOD: LISTPROVIDINGCHANNELS	209
Parameters	209



Returns	210
25.9. METHOD: LISTPROVIDINGERRATA	210
Parameters	210
Returns	210
25.10. METHOD: REMOVEPACKAGE	210
Parameters	210
Returns	210
<b>CHAPTER 26. NAMESPACE: PACKAGES.PROVIDER .....</b>	<b>212</b>
26.1. METHOD: ASSOCIATEKEY	212
Parameters	212
Returns	212
26.2. METHOD: LIST	212
Parameters	212
Returns	212
26.3. METHOD: LISTKEYS	212
Parameters	212
Returns	213
<b>CHAPTER 27. NAMESPACE: PACKAGES.SEARCH .....</b>	<b>214</b>
27.1. METHOD: ADVANCED	214
Parameters	214
Returns	214
27.2. METHOD: ADVANCEDWITHACTKEY	214
Parameters	214
Returns	214
27.3. METHOD: ADVANCEDWITHCHANNEL	215
Parameters	215
Returns	215
27.4. METHOD: NAME	216
Parameters	216
Returns	216
27.5. METHOD: NAMEANDDESCRIPTION	216
Parameters	216
Returns	216
27.6. METHOD: NAMEANDSUMMARY	217
Parameters	217
Returns	217
<b>CHAPTER 28. NAMESPACE: PREFERENCES.LOCALE .....</b>	<b>218</b>
28.1. METHOD: LISTLOCALES	218
Parameters	218
Returns	218
28.2. METHOD: LISTTIMEZONES	218
Parameters	218
Returns	218
28.3. METHOD: SETLOCALE	218
Parameters	218
Returns	218
28.4. METHOD: SETTIMEZONE	218
Parameters	218
Returns	219
<b>CHAPTER 29. NAMESPACE: PROXY .....</b>	<b>220</b>

29.1. METHOD: ACTIVATEPROXY	220
Parameters	220
Returns	220
29.2. METHOD: CREATEMONITORINGSCOUT	220
Parameters	220
Returns	220
29.3. METHOD: DEACTIVATEPROXY	220
Parameters	220
Returns	220
29.4. METHOD: ISPROXY	220
Parameters	220
Returns	220
29.5. METHOD: LISTAVAILABLEPROXYCHANNELS	221
Parameters	221
Returns	221
<b>CHAPTER 30. NAMESPACE: SATELLITE .....</b>	<b>222</b>
30.1. METHOD: GETCERTIFICATEEXPIRATIONDATE	222
Parameters	222
Returns	222
30.2. METHOD: ISMONITORINGENABLED	222
Parameters	222
Returns	222
30.3. METHOD: ISMONITORINGENABLEDBYSYSTEMID	222
Parameters	222
Returns	222
30.4. METHOD: LISTENTITLEMENTS	222
Parameters	222
Returns	222
30.5. METHOD: LISTPROXIES	223
Parameters	223
Returns	223
<b>CHAPTER 31. NAMESPACE: SCHEDULE .....</b>	<b>224</b>
31.1. METHOD: ARCHIVEACTIONS	224
Parameters	224
Returns	224
31.2. METHOD: CANCELACTIONS	224
Parameters	224
Returns	224
31.3. METHOD: DELETEACTIONS	224
Parameters	224
Returns	224
31.4. METHOD: LISTALLACTIONS	224
Parameters	225
Returns	225
31.5. METHOD: LISTARCHIVEDACTIONS	225
Parameters	225
Returns	225
31.6. METHOD: LISTCOMPLETEDACTIONS	226
Parameters	226
Returns	226
31.7. METHOD: LISTCOMPLETEDSYSTEMS	226

Parameters	226
Returns	226
31.8. METHOD: LISTFAILED ACTIONS	227
Parameters	227
Returns	227
31.9. METHOD: LISTFAILED SYSTEMS	227
Parameters	227
Returns	227
31.10. METHOD: LISTINPROGRESS ACTIONS	228
Parameters	228
Returns	228
31.11. METHOD: LISTINPROGRESS SYSTEMS	228
Parameters	228
Returns	228
31.12. METHOD: RESCHEDULE ACTIONS	229
Parameters	229
Returns	229
<b>CHAPTER 32. NAMESPACE: SYNC.MASTER</b>	<b>230</b>
32.1. METHOD: ADDTOMASTER	230
Parameters	230
Returns	230
32.2. METHOD: CREATE	230
Parameters	230
Returns	230
32.3. METHOD: DELETE	230
Parameters	230
Returns	231
32.4. METHOD: GETDEFAULTMASTER	231
Parameters	231
Returns	231
32.5. METHOD: GETMASTER	231
Parameters	231
Returns	231
32.6. METHOD: GETMASTERBYLABEL	231
Parameters	231
Returns	232
32.7. METHOD: GETMASTERORGS	232
Parameters	232
Returns	232
32.8. METHOD: GETMASTERS	232
Parameters	232
Returns	232
32.9. METHOD: MAKEDEFAULT	233
Parameters	233
Returns	233
32.10. METHOD: MAPTOLOCAL	233
Parameters	233
Returns	233
32.11. METHOD: SETCACERT	233
Parameters	233
Returns	233
32.12. METHOD: SETMASTERORGS	233

Parameters	234
Returns	234
32.13. METHOD: UNSETDEFAULTMASTER	234
Parameters	234
Returns	234
32.14. METHOD: UPDATE	234
Parameters	234
Returns	234
<b>CHAPTER 33. NAMESPACE: SYNC.SLAVE .....</b>	<b>235</b>
33.1. METHOD: CREATE	235
Parameters	235
Returns	235
33.2. METHOD: DELETE	235
Parameters	235
Returns	235
33.3. METHOD: GETALLOWEDORGS	235
Parameters	235
Returns	236
33.4. METHOD: GETSLAVE	236
Parameters	236
Returns	236
33.5. METHOD: GETSLAVEBYNAME	236
Parameters	236
Returns	236
33.6. METHOD: GETSLAVES	236
Parameters	236
Returns	237
33.7. METHOD: SETALLOWEDORGS	237
Parameters	237
Returns	237
33.8. METHOD: UPDATE	237
Parameters	237
Returns	237
<b>CHAPTER 34. NAMESPACE: SYSTEM .....</b>	<b>239</b>
34.1. METHOD: ADDENTITLEMENTS	239
Parameters	239
Returns	239
34.2. METHOD: ADDNOTE	239
Parameters	239
Returns	239
34.3. METHOD: APPLYERRATA	239
Parameters	239
Returns	240
34.4. METHOD: COMPAREPACKAGEPROFILE	240
Parameters	240
Returns	240
34.5. METHOD: COMPAREPACKAGES	240
Parameters	240
Returns	241
34.6. METHOD: CONVERTTOFLEXENTITLEMENT	241
Parameters	241

Returns	241
34.7. METHOD: CREATEPACKAGEPROFILE	241
Parameters	241
Returns	242
34.8. METHOD: CREATESYSTEMRECORD	242
Parameters	242
Returns	242
34.9. METHOD: DELETEDCUSTOMVALUES	242
Parameters	242
Returns	242
34.10. METHOD: DELETEDGUESTPROFILES	242
Parameters	242
Returns	243
34.11. METHOD: DELETENOTE	243
Parameters	243
Returns	243
34.12. METHOD: DELETENOTES	243
Parameters	243
Returns	243
34.13. METHOD: DELETEDPACKAGEPROFILE	243
Parameters	243
Returns	243
34.14. METHOD: DELETESYSTEM	243
Parameters	243
Returns	244
34.15. METHOD: DELETESYSTEM	244
Parameters	244
Returns	244
34.16. METHOD: DELETESYSTEMS	244
Parameters	244
Returns	244
34.17. METHOD: DELETETAGFROMSNAPSHOT	244
Parameters	244
Returns	244
34.18. METHOD: DOWNLOADSYSTEMID	244
Parameters	245
Returns	245
34.19. METHOD: GETCONNECTIONPATH	245
Parameters	245
Returns	245
34.20. METHOD: GETCPU	245
Parameters	245
Returns	245
34.21. METHOD: GETCUSTOMVALUES	246
Parameters	246
Returns	246
34.22. METHOD: GETDETAILS	246
Parameters	246
Returns	246
34.23. METHOD: GETDEVICES	247
Parameters	247
Returns	247
34.24. METHOD: GETDMI	248

Parameters	248
Returns	248
34.25. METHOD: GETENTITLEMENTS	248
Parameters	248
Returns	248
34.26. METHOD: GETEVENTHISTORY	248
Parameters	249
Returns	249
34.27. METHOD: GETID	249
Parameters	249
Returns	249
34.28. METHOD: GETMEMORY	249
Parameters	249
Returns	249
34.29. METHOD: GETNAME	250
Parameters	250
Returns	250
34.30. METHOD: GETNETWORK	250
Parameters	250
Returns	250
34.31. METHOD: GETNETWORKDEVICES	250
Parameters	250
Returns	250
34.32. METHOD: GETREGISTRATIONDATE	251
Parameters	251
Returns	251
34.33. METHOD: GETRELEVANTERRATA	251
Parameters	251
Returns	251
34.34. METHOD: GETRELEVANTERRATABYTYPE	252
Parameters	252
Returns	252
34.35. METHOD: GETRUNNINGKERNEL	252
Parameters	252
Returns	252
34.36. METHOD: GETSCRIPTACTIONDETAILS	253
Parameters	253
Returns	253
34.37. METHOD: GETSCRIPTRESULTS	253
Parameters	253
Returns	253
34.38. METHOD: GETSUBSCRIBEDBASECHANNEL	254
Parameters	254
Returns	254
34.39. METHOD: GETSYSTEMCURRENCYMULTIPLIERS	255
Parameters	255
Returns	255
34.40. METHOD: GETSYSTEMCURRENCYSCORES	255
Parameters	255
Returns	255
34.41. METHOD: GETUNSCHEDULEDERRATA	256
Parameters	256
Returns	256

34.42. METHOD: GETUUID	256
Parameters	256
Returns	256
34.43. METHOD: GETVARIABLES	256
Parameters	257
Returns	257
34.44. METHOD: ISNVREINSTALLED	257
Parameters	257
Returns	257
34.45. METHOD: ISNVREINSTALLED	257
Parameters	257
Returns	258
34.46. METHOD: LISTACTIVATIONKEYS	258
Parameters	258
Returns	258
34.47. METHOD: LISTACTIVESYSTEMS	258
Parameters	258
Returns	258
34.48. METHOD: LISTACTIVESYSTEMSDETAILS	258
Parameters	258
Returns	259
34.49. METHOD: LISTADMINISTRATORS	260
Parameters	260
Returns	261
34.50. METHOD: LISTALLINSTALLABLEPACKAGES	261
Parameters	261
Returns	261
34.51. METHOD: LISTBASECHANNELS	261
Parameters	261
Returns	262
34.52. METHOD: LISTCHILDCHANNELS	262
Parameters	262
Returns	262
34.53. METHOD: LISTDUPLICATESBYHOSTNAME	262
Parameters	262
Returns	262
34.54. METHOD: LISTDUPLICATESBYIP	263
Parameters	263
Returns	263
34.55. METHOD: LISTDUPLICATESBYMAC	263
Parameters	263
Returns	263
34.56. METHOD: LISTELIGIBLEFLEXGUESTS	264
Parameters	264
Returns	264
34.57. METHOD: LISTEXTRAPACKAGES	264
Parameters	264
Returns	264
34.58. METHOD: LISTFLEXGUESTS	265
Parameters	265
Returns	265
34.59. METHOD: LISTGROUPS	265
Parameters	265

Returns	265
34.60. METHOD: LISTINACTIVESYSTEMS	265
Parameters	266
Returns	266
34.61. METHOD: LISTINACTIVESYSTEMS	266
Parameters	266
Returns	266
34.62. METHOD: LISTLATESTAVAILABLEPACKAGE	266
Parameters	266
Returns	266
34.63. METHOD: LISTLATESTINSTALLABLEPACKAGES	267
Parameters	267
Returns	267
34.64. METHOD: LISTLATESTUPGRADABLEPACKAGES	267
Parameters	267
Returns	268
34.65. METHOD: LISTNEWERINSTALLEDPACKAGES	268
Parameters	268
Returns	268
34.66. METHOD: LISTNOTES	269
Parameters	269
Returns	269
34.67. METHOD: LISTOLDERINSTALLEDPACKAGES	269
Parameters	269
Returns	269
34.68. METHOD: LISTOUTOFDATESYSTEMS	270
Parameters	270
Returns	270
34.69. METHOD: LISTPACKAGEPROFILES	270
Parameters	270
Returns	270
34.70. METHOD: LISTPACKAGES	270
Parameters	270
Returns	271
34.71. METHOD: LISTPACKAGESFROMCHANNEL	271
Parameters	271
Returns	271
34.72. METHOD: LISTPHYSICALSYSTEMS	272
Parameters	272
Returns	272
34.73. METHOD: LISTSUBSCRIBABLEBASECHANNELS	272
Parameters	272
Returns	272
34.74. METHOD: LISTSUBSCRIBABLECHILDCHANNELS	272
Parameters	272
Returns	273
34.75. METHOD: LISTSUBSCRIBEDCHILDCHANNELS	273
Parameters	273
Returns	273
34.76. METHOD: LISTSYSTEMEVENTS	274
Parameters	274
Returns	274
34.77. METHOD: LISTSYSTEMEVENTS	276



Parameters	276
Returns	276
34.78. METHOD: LISTSYSTEMS	277
Parameters	277
Returns	277
34.79. METHOD: LISTSYSTEMSWITHEXTRAPACKAGES	277
Parameters	277
Returns	277
34.80. METHOD: LISTSYSTEMSWITHPACKAGE	278
Parameters	278
Returns	278
34.81. METHOD: LISTSYSTEMSWITHPACKAGE	278
Parameters	278
Returns	278
34.82. METHOD: LISTUNGROUPEDSYSTEMS	278
Parameters	278
Returns	279
34.83. METHOD: LISTUSERSYSTEMS	279
Parameters	279
Returns	279
34.84. METHOD: LISTUSERSYSTEMS	279
Parameters	279
Returns	279
34.85. METHOD: LISTVIRTUALGUESTS	279
Parameters	280
Returns	280
34.86. METHOD: LISTVIRTUALHOSTS	280
Parameters	280
Returns	280
34.87. METHOD: OBTAINREACTIVATIONKEY	280
Parameters	280
Returns	280
34.88. METHOD: OBTAINREACTIVATIONKEY	281
Parameters	281
Returns	281
34.89. METHOD: PROVISIONSYSTEM	281
Parameters	281
Returns	281
34.90. METHOD: PROVISIONSYSTEM	281
Parameters	281
Returns	281
34.91. METHOD: PROVISIONVIRTUALGUEST	281
Parameters	282
Returns	282
34.92. METHOD: PROVISIONVIRTUALGUEST	282
Parameters	282
Returns	282
34.93. METHOD: PROVISIONVIRTUALGUEST	282
Parameters	282
Returns	283
34.94. METHOD: REMOVEENTITLEMENTS	283
Parameters	283
Returns	283

34.95. METHOD: SCHEDULEAPPLYERRATA	283
Parameters	283
Returns	283
34.96. METHOD: SCHEDULEAPPLYERRATA	283
Parameters	284
Returns	284
34.97. METHOD: SCHEDULEAPPLYERRATA	284
Parameters	284
Returns	284
34.98. METHOD: SCHEDULEAPPLYERRATA	284
Parameters	284
Returns	285
34.99. METHOD: SCHEDULECERTIFICATEUPDATE	285
Parameters	285
Returns	285
34.100. METHOD: SCHEDULECERTIFICATEUPDATE	285
Parameters	285
Returns	285
34.101. METHOD: SCHEDULEGUESTACTION	285
Parameters	285
Returns	286
34.102. METHOD: SCHEDULEGUESTACTION	286
Parameters	286
Returns	286
34.103. METHOD: SCHEDULEHARDWAREREFRESH	286
Parameters	286
Returns	286
34.104. METHOD: SCHEDULEPACKAGEINSTALL	286
Parameters	286
Returns	287
34.105. METHOD: SCHEDULEPACKAGEINSTALL	287
Parameters	287
Returns	287
34.106. METHOD: SCHEDULEPACKAGEREFRESH	287
Parameters	287
Returns	287
34.107. METHOD: SCHEDULEPACKAGEREMOVE	287
Parameters	287
Returns	288
34.108. METHOD: SCHEDULEREBOOT	288
Parameters	288
Returns	288
34.109. METHOD: SCHEDULESCRIPTRUN	288
Parameters	288
Returns	288
34.110. METHOD: SCHEDULESCRIPTRUN	288
Parameters	289
Returns	289
34.111. METHOD: SCHEDULESYNCPACKAGESWITHSYSTEM	289
Parameters	289
Returns	289
34.112. METHOD: SEARCHBYNAME	289
Parameters	289

Returns	290
34.113. METHOD: SETBASECHANNEL	290
Parameters	290
Returns	290
34.114. METHOD: SETBASECHANNEL	290
Parameters	290
Returns	290
34.115. METHOD: SETCHILDCHANNELS	290
Parameters	290
Returns	291
34.116. METHOD: SETCUSTOMVALUES	291
Parameters	291
Returns	291
34.117. METHOD: SETDETAILS	291
Parameters	291
Returns	292
34.118. METHOD: SETGROUPMEMBERSHIP	292
Parameters	292
Returns	292
34.119. METHOD: SETGUESTCPUS	292
Parameters	292
Returns	292
34.120. METHOD: SETGUESTMEMORY	292
Parameters	293
Returns	293
34.121. METHOD: SETLOCKSTATUS	293
Parameters	293
Returns	293
34.122. METHOD: SETPRIMARYINTERFACE	293
Parameters	293
Returns	293
34.123. METHOD: SETPROFILENAME	293
Parameters	293
Returns	294
34.124. METHOD: SETVARIABLES	294
Parameters	294
Returns	294
34.125. METHOD: TAGLATESTSNAPSHOT	294
Parameters	294
Returns	294
34.126. METHOD: UNENTITLE	294
Parameters	294
Returns	295
34.127. METHOD: UPGRADEENTITLEMENT	295
Parameters	295
Returns	295
34.128. METHOD: WHOREGISTERED	295
Parameters	295
Returns	295
<b>CHAPTER 35. NAMESPACE: SYSTEM.CONFIG .....</b>	<b>296</b>
35.1. METHOD: ADDCHANNELS	296
Parameters	296

Returns	296
35.2. METHOD: CREATEORUPDATEPATH	296
Parameters	296
Returns	297
35.3. METHOD: CREATEORUPDATESYMLINK	298
Parameters	298
Returns	298
35.4. METHOD: DELETEFILES	299
Parameters	299
Returns	300
35.5. METHOD: DEPLOYALL	300
Parameters	300
Returns	300
35.6. METHOD: LISTCHANNELS	300
Parameters	300
Returns	300
35.7. METHOD: LISTFILES	301
Parameters	301
Returns	301
35.8. METHOD: LOOKUPFILEINFO	302
Parameters	302
Returns	302
35.9. METHOD: REMOVECHANNELS	303
Parameters	303
Returns	303
35.10. METHOD: SETCHANNELS	303
Parameters	303
Returns	303
<b>CHAPTER 36. NAMESPACE: SYSTEM.CRASH .....</b>	<b>304</b>
36.1. METHOD: CREATECRASHNOTE	304
Parameters	304
Returns	304
36.2. METHOD: DELETECRASH	304
Parameters	304
Returns	304
36.3. METHOD: DELETECRASHNOTE	304
Parameters	304
Returns	304
36.4. METHOD: GETCRASHCOUNTINFO	304
Parameters	305
Returns	305
36.5. METHOD: GETCRASHFILE	305
Parameters	305
Returns	305
36.6. METHOD: GETCRASHFILEURL	305
Parameters	305
Returns	305
36.7. METHOD: GETCRASHNOTESFORCRASH	305
Parameters	305
Returns	305
36.8. METHOD: GETCRASHOVERVIEW	306
Parameters	306

Returns	306
36.9. METHOD: GETCRASHESBYUUID	306
Parameters	306
Returns	306
36.10. METHOD: LISTSYSTEMCRASHFILES	307
Parameters	307
Returns	307
36.11. METHOD: LISTSYSTEMCRASHES	307
Parameters	307
Returns	307
<b>CHAPTER 37. NAMESPACE: SYSTEM.CUSTOMINFO .....</b>	<b>309</b>
37.1. METHOD: CREATEKEY	309
Parameters	309
Returns	309
37.2. METHOD: DELETEKEY	309
Parameters	309
Returns	309
37.3. METHOD: LISTALLKEYS	309
Parameters	309
Returns	309
37.4. METHOD: UPDATEKEY	310
Parameters	310
Returns	310
<b>CHAPTER 38. NAMESPACE: SYSTEM.PROVISIONING.SNAPSHOT .....</b>	<b>311</b>
38.1. METHOD: ADDTAGTOSNAPSHOT	311
Parameters	311
Returns	311
38.2. METHOD: DELETESNAPSHOT	311
Parameters	311
Returns	311
38.3. METHOD: DELETESNAPSHOTS	311
Parameters	311
Returns	312
38.4. METHOD: DELETESNAPSHOTS	312
Parameters	312
Returns	312
38.5. METHOD: LISTSNAPSHOTCONFIGFILES	312
Parameters	312
Returns	312
38.6. METHOD: LISTSNAPSHOTPACKAGES	313
Parameters	313
Returns	313
38.7. METHOD: LISTSNAPSHOTS	314
Parameters	314
Returns	314
38.8. METHOD: ROLLBACKTOSNAPSHOT	315
Parameters	315
Returns	315
38.9. METHOD: ROLLBACKTOTAG	315
Parameters	315
Returns	315

38.10. METHOD: ROLLBACKTOTAG	315
Parameters	316
Returns	316
<b>CHAPTER 39. NAMESPACE: SYSTEM.SCAP .....</b>	<b>317</b>
39.1. METHOD: DELETEXCCDFSCAN	317
Parameters	317
Returns	317
39.2. METHOD: GETXCCDFSCANDetails	317
Parameters	317
Returns	317
39.3. METHOD: GETXCCDFSCANRULERESULTS	318
Parameters	318
Returns	318
39.4. METHOD: LISTXCCDFSCANS	318
Parameters	318
Returns	318
39.5. METHOD: SCHEDULEXCCDFSCAN	318
Parameters	319
Returns	319
39.6. METHOD: SCHEDULEXCCDFSCAN	319
Parameters	319
Returns	319
39.7. METHOD: SCHEDULEXCCDFSCAN	319
Parameters	319
Returns	319
39.8. METHOD: SCHEDULEXCCDFSCAN	320
Parameters	320
Returns	320
<b>CHAPTER 40. NAMESPACE: SYSTEM.SEARCH .....</b>	<b>321</b>
40.1. METHOD: DEVIDEDESCRIPTION	321
Parameters	321
Returns	321
40.2. METHOD: DEVIDEDRIVER	321
Parameters	321
Returns	321
40.3. METHOD: DEVIDEID	322
Parameters	322
Returns	322
40.4. METHOD: DEVIDEVENDORID	322
Parameters	322
Returns	323
40.5. METHOD: HOSTNAME	323
Parameters	323
Returns	323
40.6. METHOD: IP	324
Parameters	324
Returns	324
40.7. METHOD: NAMEANDDESCRIPTION	324
Parameters	324
Returns	324
40.8. METHOD: UUID	325

Parameters	325
Returns	325
<b>CHAPTER 41. NAMESPACE: SYSTEMGROUP .....</b>	<b>326</b>
41.1. METHOD: ADDORREMOVEADMINS	326
Parameters	326
Returns	326
41.2. METHOD: ADDORREMOVESYSTEMS	326
Parameters	326
Returns	326
41.3. METHOD: CREATE	326
Parameters	326
Returns	327
41.4. METHOD: DELETE	327
Parameters	327
Returns	327
41.5. METHOD: GETDETAILS	327
Parameters	327
Returns	327
41.6. METHOD: GETDETAILS	327
Parameters	328
Returns	328
41.7. METHOD: LISTACTIVESYSTEMSINGROUP	328
Parameters	328
Returns	328
41.8. METHOD: LISTADMINISTRATORS	328
Parameters	328
Returns	328
41.9. METHOD: LISTALLGROUPS	329
Parameters	329
Returns	329
41.10. METHOD: LISTGROUPSWITHNOASSOCIATEDADMINS	329
Parameters	329
Returns	329
41.11. METHOD: LISTINACTIVESYSTEMSINGROUP	330
Parameters	330
Returns	330
41.12. METHOD: LISTINACTIVESYSTEMSINGROUP	330
Parameters	330
Returns	330
41.13. METHOD: LISTSYSTEMS	330
Parameters	330
Returns	330
41.14. METHOD: LISTSYSTEMSMINIMAL	331
Parameters	331
Returns	331
41.15. METHOD: SCHEDULEAPPLYERRATATOACTIVE	332
Parameters	332
Returns	332
41.16. METHOD: SCHEDULEAPPLYERRATATOACTIVE	332
Parameters	332
Returns	332
41.17. METHOD: UPDATE	332

Parameters	333
Returns	333
<b>CHAPTER 42. NAMESPACE: USER.EXTERNAL .....</b>	<b>334</b>
42.1. METHOD: CREATEEXTERNALGROUPTOROLEMAP	334
Parameters	334
Returns	334
42.2. METHOD: CREATEEXTERNALGROUPTOSYSTEMGROUPMAP	334
Parameters	334
Returns	334
42.3. METHOD: DELETEEXTERNALGROUPTOROLEMAP	335
Parameters	335
Returns	335
42.4. METHOD: DELETEEXTERNALGROUPTOSYSTEMGROUPMAP	335
Parameters	335
Returns	335
42.5. METHOD: GETDEFAULTORG	335
Parameters	335
Returns	335
42.6. METHOD: GETEXTERNALGROUPTOROLEMAP	335
Parameters	335
Returns	336
42.7. METHOD: GETEXTERNALGROUPTOSYSTEMGROUPMAP	336
Parameters	336
Returns	336
42.8. METHOD: GETKEEPTEMPORARYROLES	336
Parameters	336
Returns	336
42.9. METHOD: GETUSEORGUNIT	336
Parameters	336
Returns	337
42.10. METHOD: LISTEXTERNALGROUPTOROLEMAPS	337
Parameters	337
Returns	337
42.11. METHOD: LISTEXTERNALGROUPTOSYSTEMGROUPMAPS	337
Parameters	337
Returns	337
42.12. METHOD: SETDEFAULTORG	337
Parameters	337
Returns	338
42.13. METHOD: SETEXTERNALGROUPPROLES	338
Parameters	338
Returns	338
42.14. METHOD: SETEXTERNALGROUPSYSTEMGROUPS	338
Parameters	338
Returns	338
42.15. METHOD: SETKEEPTEMPORARYROLES	338
Parameters	338
Returns	339
42.16. METHOD: SETUSEORGUNIT	339
Parameters	339
Returns	339



<b>CHAPTER 43. NAMESPACE: USER .....</b>	<b>340</b>
43.1. METHOD: ADDASSIGNEDSYSTEMGROUP	340
Parameters	340
Returns	340
43.2. METHOD: ADDASSIGNEDSYSTEMGROUPS	340
Parameters	340
Returns	340
43.3. METHOD: ADDDEFAULTSYSTEMGROUP	340
Parameters	340
Returns	341
43.4. METHOD: ADDDEFAULTSYSTEMGROUPS	341
Parameters	341
Returns	341
43.5. METHOD: ADDROLE	341
Parameters	341
Returns	341
43.6. METHOD: CREATE	341
Parameters	341
Returns	342
43.7. METHOD: CREATE	342
Parameters	342
Returns	342
43.8. METHOD: DELETE	342
Parameters	342
Returns	342
43.9. METHOD: DISABLE	342
Parameters	342
Returns	343
43.10. METHOD: ENABLE	343
Parameters	343
Returns	343
43.11. METHOD: GETCREATEDDEFAULTSYSTEMGROUP	343
Parameters	343
Returns	343
43.12. METHOD: GETDETAILS	343
Parameters	343
Returns	343
43.13. METHOD: GETLOGGEDINTIME	344
Parameters	344
Returns	344
43.14. METHOD: LISTASSIGNABLEROLES	344
Parameters	344
Returns	344
43.15. METHOD: LISTASSIGNEDSYSTEMGROUPS	344
Parameters	344
Returns	345
43.16. METHOD: LISTDEFAULTSYSTEMGROUPS	345
Parameters	345
Returns	345
43.17. METHOD: LISTROLES	345
Parameters	345
Returns	345
43.18. METHOD: LISTUSERS	346

Parameters	346
Returns	346
43.19. METHOD: REMOVEASSIGNEDSYSTEMGROUP	346
Parameters	346
Returns	346
43.20. METHOD: REMOVEASSIGNEDSYSTEMGROUPS	346
Parameters	346
Returns	347
43.21. METHOD: REMOVEDEFAULTSYSTEMGROUP	347
Parameters	347
Returns	347
43.22. METHOD: REMOVEDEFAULTSYSTEMGROUPS	347
Parameters	347
Returns	347
43.23. METHOD: REMOVEROLE	347
Parameters	347
Returns	348
43.24. METHOD: SETCREATEDDEFAULTSYSTEMGROUP	348
Parameters	348
Returns	348
43.25. METHOD: SETDETAILS	348
Parameters	348
Returns	348
43.26. METHOD: SETERRATANOTIFICATIONS	349
Parameters	349
Returns	349
43.27. METHOD: SETREADONLY	349
Parameters	349
Returns	349
43.28. METHOD: USEPAMAUTHENTICATION	349
Parameters	349
Returns	349
<b>APPENDIX A. REVISION HISTORY</b> .....	<b>350</b>



## CHAPTER 1. INTRODUCTION TO RED HAT SATELLITE

Red Hat Satellite provides organizations with the benefits of Red Hat Network without the need for public Internet access for servers or client systems. Red Hat Satellite also provides the following benefits:

- You have complete control and privacy over package management and server maintenance within your own networks.
- You can store System Profiles on a Satellite server, which connects to the Red Hat Network website using a local web server.
- You can perform package management tasks, including errata updates, through the local area network.

This gives Red Hat Network customers the greatest flexibility and power to keep servers secure and up-to-date.

### 1.1. THE RED HAT SATELLITE API

Red Hat Satellite includes an application programming interface (API), which offers software developers and system administrators control over their Satellite servers outside of the standard web and command line interfaces. The API is useful for developers and administrators who aim to integrate the functionality of a Red Hat Satellite with custom scripts or external applications that access the API via XML remote procedure call (RPC) protocol, or XML-RPC.

This guide aims to provide developers and administrators with instructions and examples to help harness the functionality of their Red Hat Satellite through the XML-RPM protocol.

### 1.2. USING XML-RPC WITH THE RED HAT SATELLITE API

XML-RPC uses HTTP to send an XML-encoded request to a server. The request contains the following:

#### Namespace

A namespace is a grouping of methods based upon a particular function, object or resource. For example, the `auth` namespace groups authentication function, or the `errata` namespace groups function that control errata.

#### Method

A method represents a certain action. Each method controls a specific function of the Red Hat Satellite. For example, the `login` method in the `auth` namespace logs a user into Red Hat Satellite and returns a session key.

#### Parameter

A parameter is a piece of input to control specific aspects of a method. For example, the `login` method in the `auth` namespace requires `username` and `password` parameters to specify the login details of a certain user. The `login` method also accepts an optional `duration` parameter to specify the length of time until the user session expires..

The XML-encoded request usually appears in the following structure:

```
<methodCall>
```

```

<methodName>namespace.method</methodName>
<params>
  <param>
    <value><name>parameter</name></value>
  </param>
  <param>
    <value><name>parameter</name></value>
  </param>
  ...
</params>
</methodCall>

```

For example, use the following XML-RPC request to login to the API and request a session key for authentication:

```

<methodCall>
  <methodName>auth.login</methodName>
  <params>
    <param>
      <value><username>admin</username></value>
    </param>
    <param>
      <value><password>p@55w0rd!</password></value>
    </param>
  </params>
</methodCall>

```

In this example, the request sends the **username** and **password** as a parameters to the **login** method, which is a part of the **auth** namespace. The Satellite server returns the following response:

```

<methodResponse>
  <params>
    <param>
      <value><sessionKey>83d8b35f</sessionKey></value>
    </param>
  </params>
</methodResponse>

```

The response contains a returned parameter for **sessionKey**, which is a key used to authenticate most of the API methods.

Most programming languages include XML-RPC modules and libraries that automatically parse the desired method and parameters into the XML-encoded request and parse the resulting response as returned variable. For examples, see [Chapter 2, Examples](#).

For more information about XML-RPC, see <http://www.xmlrpc.com/>.

### 1.3. USING THE READ-ONLY API

Red Hat Satellite 5.7 features a read-only API that is designed to provide a safe way of auditing and generating reports of the contents of the Satellite server. Read-only users cannot log in to the Satellite web UI or perform other actions that can affect the functionality of the Satellite. For example, read-only users cannot make back-end calls or use API calls that modify content.

Read-only API users can only use non-destructive API calls, such as `org.listUsers`; in short, they

are denied access to any API call that does not start with **list**, **is**, or **get**. In addition, read-only users require the same role specifications as normal users to gain access to the API calls associated with those roles. This leads to the concept where a read-only channel administrator can make API calls such as **channel.listAllChannels** but cannot call **org.listUsers** because the organization administrator rights are not available.

### 1.3.1. Creating a Read-only User

Use the same procedure to create a read-only user as for any other type of user, but ensure you select the **Read only API user** as part of the process.

#### Procedure 1.1. To Create a Read-only API User:

1. Click the **Users** tab on the main menu, and then click **Create User**.
2. Complete the required details, and select **Read-only API User**.
3. Click **Create Login**.

## CHAPTER 2. EXAMPLES

The following sections demonstrate Red Hat Satellite API usage using different programming languages and their respective XML-RPC requests.

### 2.1. PYTHON EXAMPLES

The following example demonstrates the `user.listUsers` call. The example prints the name of each group.

```
#!/usr/bin/python
import xmlrpclib

SATELLITE_URL = "http://satellite.example.com/rpc/api"
SATELLITE_LOGIN = "username"
SATELLITE_PASSWORD = "password"

client = xmlrpclib.Server(SATELLITE_URL, verbose=0)

key = client.auth.login(SATELLITE_LOGIN, SATELLITE_PASSWORD)
list = client.user.list_users(key)
for user in list:
    print user.get('login')

client.auth.logout(key)
```

The following example shows how to use date-time parameters. This code schedules the installation of package `rhnlb-2.5.22.9.el6.noarch` to system with id `1000000001`.

```
#!/usr/bin/python
from datetime import datetime
import time
import xmlrpclib

SATELLITE_URL = "http://satellite.example.com/rpc/api"
SATELLITE_LOGIN = "username"
SATELLITE_PASSWORD = "password"

client = xmlrpclib.Server(SATELLITE_URL, verbose=0)

key = client.auth.login(SATELLITE_LOGIN, SATELLITE_PASSWORD)
package_list = client.packages.findByName(key, 'rhnlb', '2.5.22',
    '9.el6', '', 'noarch')
today = datetime.today()
earliest_occurrence = xmlrpclib.DateTime(today)
client.system.schedulePackageInstall(key, 1000000001, package_list[0]
    ['id'], earliest_occurrence)

client.auth.logout(key)
```

The following example show how to check channels for their last synchronization:

```
#!/usr/bin/python
```

```
import xmlrpclib

SATELLITE_URL = "http://sat57.example.com/rpc/api"
SATELLITE_LOGIN = "admin"
SATELLITE_PASSWORD = "Redhat123"

client = xmlrpclib.Server(SATELLITE_URL, verbose=0)

key = client.auth.login(SATELLITE_LOGIN, SATELLITE_PASSWORD)
list = client.channel.listAllChannels(key)

for item in list:
    details = client.channel.software.getDetails(key, item['label'])
    print item
    print details
    print "Name: " + details['name']
    try:
        print "Last Sync: " + details['yumrepo_last_sync']
    except:
        print "Last Sync: Never"

client.auth.logout(key)
```

## 2.2. PERL EXAMPLE

This example uses the `system.listUserSystems` call to retrieve a list of systems a user can access. The example prints the name of each system. The `Frontier::Client` Perl module is found in the `perl-Frontier-RPC` RPM file.

```
#!/usr/bin/perl
use Frontier::Client;

my $HOST = 'satellite.example.com';
my $user = 'username';
my $pass = 'password';

my $client = new Frontier::Client(url => "http://$HOST/rpc/api");
my $session = $client->call('auth.login',$user, $pass);

my $systems = $client->call('system.listUserSystems', $session);
foreach my $system (@$systems) {
    print $system->{'name'}."\n";
}
$client->call('auth.logout', $session);
```

## 2.3. RUBY EXAMPLE

The following example demonstrates the `channel.listAllChannels` API call. The example prints a list of channel labels.

```
#!/usr/bin/env ruby
require "xmlrpc/client"
```



```
@SATELLITE_URL = "http://satellite.example.com/rpc/api"
@SATELLITE_LOGIN = "username"
@SATELLITE_PASSWORD = "password"

@client = XMLRPC::Client.new2(@SATELLITE_URL)

@key = @client.call('auth.login', @SATELLITE_LOGIN, @SATELLITE_PASSWORD)
channels = @client.call('channel.listAllChannels', @key)
for channel in channels do
  p channel["label"]
end

@client.call('auth.logout', @key)
```

PART I. METHODS

## CHAPTER 3. NAMESPACE: ACTIONCHAIN

Provides the namespace for the Action Chain methods.

### 3.1. METHOD: ADDCONFIGURATIONDEPLOYMENT

Adds an action to deploy a configuration file to an Action Chain.

#### Parameters

- string `sessionKey` - Session token, issued at login
- string `chainLabel` - Label of the chain
- int `systemId` - System ID
- array
  - int - Revision ID

#### Returns

- int - 1 on success, exception thrown otherwise.

### 3.2. METHOD: ADDPACKAGEINSTALL

Adds package installation action to an Action Chain.

#### Parameters

- string `sessionKey` - Session token, issued at login
- int `serverId` - System ID
- array
  - int - Package ID
- string `chainLabel`

#### Returns

- int - 1 on success, exception thrown otherwise.

### 3.3. METHOD: ADDPACKAGEREMOVAL

Adds an action to remove installed packages on the system to an Action Chain.

#### Parameters

- string `sessionKey` - Session token, issued at login
- int `serverId` - System ID
- array

- int - Package ID
- string `chainLabel` - Label of the chain

### Returns

- int `actionId` - The action id of the scheduled action or exception

## 3.4. METHOD: ADDPACKAGEUPGRADE

Adds an action to upgrade installed packages on the system to an Action Chain.

### Parameters

- string `sessionKey` - Session token, issued at login
- int `serverId` - System ID
- array
  - int `packageId`
- string `chainLabel` - Label of the chain

### Returns

- int `actionId` - The action id of the scheduled action or exception

## 3.5. METHOD: ADDPACKAGEVERIFY

Adds an action to verify installed packages on the system to an Action Chain.

### Parameters

- string `sessionKey` - Session token, issued at login
- int `serverId` - System ID
- array
  - int `packageId`
- string `chainLabel` - Label of the chain

### Returns

- int - 1 on success, exception thrown otherwise.

## 3.6. METHOD: ADDSCRIPTRUN

Add an action to run a script to an Action Chain. NOTE: The script body must be Base64 encoded!

### Parameters

- string `sessionKey` - Session token, issued at login

- `int serverId` - System ID
- `string chainLabel` - Label of the chain
- `string uid` - User ID on the particular system
- `string gid` - Group ID on the particular system
- `int timeout` - Timeout
- `string scriptBodyBase64` - Base64 encoded script body

### Returns

- `int actionId` - The id of the action or throw an exception

## 3.7. METHOD: ADDSYSTEMREBOOT

Add system reboot to an Action Chain.

### Parameters

- `string sessionKey` - Session token, issued at login
- `int serverId`
- `string chainLabel` - Label of the chain

### Returns

- `int actionId` - The action id of the scheduled action

## 3.8. METHOD: CREATECHAIN

Create an Action Chain.

### Parameters

- `string sessionKey` - Session token, issued at login
- `string chainLabel` - Label of the chain

### Returns

- `int actionId` - The ID of the created action chain

## 3.9. METHOD: DELETECHAIN

Delete action chain by label.

### Parameters

- `string sessionKey` - Session token, issued at login
- `string chainLabel` - Label of the chain

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 3.10. METHOD: LISTCHAINACTIONS

List all actions in the particular Action Chain.

### Parameters

- `string sessionKey` - Session token, issued at login
- `string chainLabel` - Label of the chain

### Returns

- `array`
  - `struct entry`
    - `int id` - Action ID
    - `string label` - Label of an Action
    - `string created` - Created date/time
    - `string earliest` - Earliest scheduled date/time
    - `string type` - Type of the action
    - `string modified` - Modified date/time
    - `string cuid` - Creator UID

## 3.11. METHOD: LISTCHAINS

List currently available action chains.

### Parameters

- `string sessionKey` - Session token, issued at login

### Returns

- `array`
  - `struct chain`
    - `string label` - Label of an Action Chain
    - `string entrycount` - Number of entries in the Action Chain

## 3.12. METHOD: REMOVEACTION

Remove an action from an Action Chain.

## Parameters

- string `sessionKey` - Session token, issued at login
- string `chainLabel` - Label of the chain
- int `actionId` - Action ID

## Returns

- int - 1 on success, exception thrown otherwise.

## 3.13. METHOD: RENAMECHAIN

Rename an Action Chain.

### Parameters

- string `sessionKey` - Session token, issued at login
- string `previousLabel` - Previous chain label
- string `newLabel` - New chain label

### Returns

- int - 1 on success, exception thrown otherwise.

## 3.14. METHOD: SCHEDULECHAIN

Schedule the Action Chain so that its actions will actually occur.

### Parameters

- string `sessionKey` - Session token, issued at login
- string `chainLabel` - Label of the chain
- `dateTime.iso8601` - Earliest date

### Returns

- int - 1 on success, exception thrown otherwise.

## CHAPTER 4. NAMESPACE: ACTIVATIONKEY

Contains methods to access common activation key functions available from the web interface.

### 4.1. METHOD: ADDCHILDCHANNELS

Add child channels to an activation key.

#### Parameters

- `string sessionKey`
- `string key`
- `array`
  - `string childChannelLabel`

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 4.2. METHOD: ADDCONFIGCHANNELS

Given a list of activation keys and configuration channels, this method adds given configuration channels to either the top or the bottom (whichever you specify) of an activation key's configuration channels list. The ordering of the configuration channels provided in the add list is maintained while adding. If one of the configuration channels in the 'add' list already exists in an activation key, the configuration channel will be re-ranked to the appropriate place.

#### Parameters

- `string sessionKey`
- `array`
  - `string activationKey`
- `array`
  - `string` - List of configuration channel labels in the ranked order.
- `boolean addToTop`
  - `true` - To prepend the given channels to the beginning of the activation key's config channel list
  - `false` - To append the given channels to the end of the activation key's config channel list

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 4.3. METHOD: ADDENTITLEMENTS



Add entitlements to an activation key. Currently only add-on entitlements are permitted.  
(monitoring\_entitled, provisioning\_entitled, virtualization\_host, virtualization\_host\_platform)

### Parameters

- string `sessionKey`
- string `key`
- array
  - string - entitlement label
    - `monitoring_entitled`
    - `provisioning_entitled`
    - `virtualization_host`
    - `virtualization_host_platform`

### Returns

- int - 1 on success, exception thrown otherwise.

## 4.4. METHOD: ADDPACKAGENAMES

Add packages to an activation key using package name only.

Deprecated - being replaced by `addPackages(string sessionKey, string key, array[packages])`

### Parameters

- string `sessionKey`
- string `key`
- array
  - string `packageName`

### Returns

- int - 1 on success, exception thrown otherwise.

Available since: 10.2

## 4.5. METHOD: ADDPACKAGES

Add packages to an activation key.

### Parameters

- string `sessionKey`
- string `key`

- array
  - struct **packages**
    - string **name** - Package name
    - string **arch** - Arch label - Optional

## Returns

- int - 1 on success, exception thrown otherwise.

## 4.6. METHOD: ADDSERVERGROUPS

Add server groups to an activation key.

### Parameters

- string **sessionKey**
- string **key**
- array
  - int **serverGroupId**

## Returns

- int - 1 on success, exception thrown otherwise.

## 4.7. METHOD: CHECKCONFIGDEPLOYMENT

Check configuration file deployment status for the activation key specified.

### Parameters

- string **sessionKey**
- string **key**

## Returns

- 1 if enabled, 0 if disabled, exception thrown otherwise.

## 4.8. METHOD: CLONE

Clone an existing activation key.

### Parameters

- string **sessionKey**
- string **key** - Key to be cloned.
- string **cloneDescription** - Description of the cloned key.

## Returns

- string - The new activation key.

## 4.9. METHOD: CREATE

Create a new activation key. The activation key parameter passed in will be prefixed with the organization ID, and this value will be returned from the create call. Eg. If the caller passes in the key "foo" and belong to an organization with the ID 100, the actual activation key will be "100-foo". This call allows for the setting of a usage limit on this activation key. If unlimited usage is desired see the similarly named API method with no usage limit argument.

## Parameters

- string `sessionKey`
- string `key` - Leave empty to have new key autogenerated.
- string `description`
- string `baseChannelLabel` - Leave empty to accept default.
- int `usageLimit` - If unlimited usage is desired, use the create API that does not include the parameter.
- array
  - string - Add-on entitlement label to associate with the key.
    - `monitoring_entitled`
    - `provisioning_entitled`
    - `virtualization_host`
    - `virtualization_host_platform`
- boolean `universalDefault`

## Returns

- string - The new activation key.

## 4.10. METHOD: CREATE

Create a new activation key with unlimited usage. The activation key parameter passed in will be prefixed with the organization ID, and this value will be returned from the create call. Eg. If the caller passes in the key "foo" and belong to an organization with the ID 100, the actual activation key will be "100-foo".

## Parameters

- string `sessionKey`
- string `key` - Leave empty to have new key autogenerated.

- string **description**
- string **baseChannelLabel** - Leave empty to accept default.
- array
  - string - Add-on entitlement label to associate with the key.
    - **monitoring\_entitled**
    - **provisioning\_entitled**
    - **virtualization\_host**
    - **virtualization\_host\_platform**
- boolean **universalDefault**

### Returns

- string - The new activation key.

## 4.11. METHOD: DELETE

Delete an activation key.

### Parameters

- string **sessionKey**
- string **key**

### Returns

- int - 1 on success, exception thrown otherwise.

## 4.12. METHOD: DISABLECONFIGDEPLOYMENT

Disable configuration file deployment for the specified activation key.

### Parameters

- string **sessionKey**
- string **key**

### Returns

- int - 1 on success, exception thrown otherwise.

## 4.13. METHOD: ENABLECONFIGDEPLOYMENT

Enable configuration file deployment for the specified activation key.

### Parameters

- string `sessionKey`
- string `key`

### Returns

- int - 1 on success, exception thrown otherwise.

## 4.14. METHOD: GETDETAILS

Lookup an activation key's details.

### Parameters

- string `sessionKey`
- string `key`

### Returns

- struct `activation key`
  - string `key`
  - string `description`
  - int `usage_limit`
  - string `base_channel_label`
  - array `child_channel_labels`
    - string `childChannelLabel`
  - array `entitlements`
    - string `entitlementLabel`
  - array `server_group_ids`
    - string `serverGroupId`
  - array `package_names`
    - string `packageName` - (deprecated by packages)
  - array `packages`
    - struct `package`
      - string `name` - `packageName`
      - string `arch` - `archLabel` - optional
  - boolean `universal_default`

- `boolean disabled`

Available since: 10.2

## 4.15. METHOD: LISTACTIVATEDSYSTEMS

List the systems activated with the key provided.

### Parameters

- `string sessionKey`
- `string key`

### Returns

- `array`
  - `struct system` structure
    - `int id` - System id
    - `string hostname`
    - `dateTime.iso8601 last_checkin` - Last time server successfully checked in

## 4.16. METHOD: LISTACTIVATIONKEYS

List activation keys that are visible to the user.

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct activation` key
    - `string key`
    - `string description`
    - `int usage_limit`
    - `string base_channel_label`
    - `array child_channel_labels`
      - `string childChannelLabel`
    - `array entitlements`
      - `string entitlementLabel`

- array `server_group_ids`
  - string `serverGroupId`
- array `package_names`
  - string `packageName` - (deprecated by packages)
- array `packages`
  - struct `package`
    - string `name` - `packageName`
    - string `arch` - `archLabel` - optional
- boolean `universal_default`
- boolean `disabled`

Available since: 10.2

## 4.17. METHOD: LISTCONFIGCHANNELS

List configuration channels associated to an activation key.

### Parameters

- string `sessionKey`
- string `key`

### Returns

array

struct `Configuration Channel` information

int `id`

int `orgId`

string `label`

string `name`

string `description`

struct `configChannelType`

struct `Configuration Channel Type` information

- int `id`
- string `label`
- string `name`

- `int priority`

## 4.18. METHOD: REMOVECHILDCHANNELS

Remove child channels from an activation key.

### Parameters

- `string sessionKey`
- `string key`
- `array`
  - `string childChannelLabel`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 4.19. METHOD: REMOVECONFIGCHANNELS

Remove configuration channels from the given activation keys.

### Parameters

- `string sessionKey`
- `array`
  - `string activationKey`
- `array`
  - `string configChannelLabel`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 4.20. METHOD: REMOVEENTITLEMENTS

Remove entitlements (by label) from an activation key. Currently only add-on entitlements are permitted. (monitoring\_entitled, provisioning\_entitled, virtualization\_host, virtualization\_host\_platform)

### Parameters

- `string sessionKey`
- `string key`
- `array`
  - `string` - entitlement label



- `monitoring_entitled`
- `provisioning_entitled`
- `virtualization_host`
- `virtualization_host_platform`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 4.21. METHOD: REMOVEPACKAGENAMES

Remove package names from an activation key.

Deprecated - being replaced by `removePackages(string sessionKey, string key, array[packages])`

### Parameters

- `string sessionKey`
- `string key`
- `array`
  - `string packageName`

### Returns

- `int` - 1 on success, exception thrown otherwise.

Available since: 10.2

## 4.22. METHOD: REMOVEPACKAGES

Remove package names from an activation key.

### Parameters

- `string sessionKey`
- `string key`
- `array`
  - `struct packages`
    - `string name` - Package name
    - `string arch` - Arch label - Optional

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 4.23. METHOD: REMOVESERVERGROUPS

Remove server groups from an activation key.

### Parameters

- string `sessionKey`
- string `key`
- array
  - int `serverGroupId`

### Returns

- int - 1 on success, exception thrown otherwise.

## 4.24. METHOD: SETCONFIGCHANNELS

Replace the existing set of configuration channels on the given activation keys. Channels are ranked by their order in the array.

### Parameters

- string `sessionKey`
- array
  - string `activationKey`
- array
  - string `configChannelLabel`

### Returns

- int - 1 on success, exception thrown otherwise.

## 4.25. METHOD: SETDETAILS

Update the details of an activation key.

### Parameters

- string `sessionKey`
- string `key`
- struct `activation key`
  - string `description` - optional
  - string `base_channel_label` - optional - to set default base channel set to empty string or 'none'

- `int usage_limit` - optional
- `boolean unlimited_usage_limit` - Set true for unlimited usage and to override `usage_limit`
- `boolean universal_default` - optional
- `boolean disabled` - optional

## Returns

- `int` - 1 on success, exception thrown otherwise.

## CHAPTER 5. NAMESPACE: API

Methods providing information about the API.

### 5.1. METHOD: GETAPICALLLIST

Lists all available api calls grouped by namespace

#### Parameters

- `string sessionKey`

#### Returns

- `struct method_info`
  - `string name` - method name
  - `string parameters` - method parameters
  - `string exceptions` - method exceptions
  - `string return` - method return type

### 5.2. METHOD: GETAPINAMESPACECALLLIST

Lists all available api calls for the specified namespace

#### Parameters

- `string sessionKey`
- `string namespace`

#### Returns

- `struct method_info`
  - `string name` - method name
  - `string parameters` - method parameters
  - `string exceptions` - method exceptions
  - `string return` - method return type

### 5.3. METHOD: GETAPINAMESPACES

Lists available API namespaces

#### Parameters

- `string sessionKey`

## Returns

- `struct namespace`
  - `string namespace` - API namespace
  - `string handler` - API Handler

## 5.4. METHOD: GETVERSION

Returns the version of the API. Since Spacewalk 0.4 (Satellite 5.3) it is no more related to server version.

### Parameters

### Returns

- `string`

## 5.5. METHOD: SYSTEMVERSION

Returns the server version.

### Parameters

### Returns

- `string`

## CHAPTER 6. NAMESPACE: AUTH

This namespace provides methods to authenticate with the system's management server.

### 6.1. METHOD: LOGIN

Login using a username and password. Returns the session key used by most other API methods.

#### Parameters

- `string username`
- `string password`

#### Returns

- `string sessionKey`

### 6.2. METHOD: LOGIN

Login using a username and password. Returns the session key used by other methods.

#### Parameters

- `string username`
- `string password`
- `int duration` - Length of session.

#### Returns

- `string sessionKey`

### 6.3. METHOD: LOGOUT

Logout the user with the given session key.

#### Parameters

- `string sessionKey`

#### Returns

- `int` - 1 on success, exception thrown otherwise.

## CHAPTER 7. NAMESPACE: CHANNEL

Provides method to get back a list of Software Channels.

### 7.1. METHOD: LISTALLCHANNELS

List all software channels that the user's organization is entitled to.

#### Parameters

- `string sessionKey`

#### Returns

- `array`
  - `struct channelInfo`
    - `int id`
    - `string label`
    - `string name`
    - `string provider_name`
    - `int packages`
    - `int systems`
    - `string arch_name`

### 7.2. METHOD: LISTMYCHANNELS

List all software channels that belong to the user's organization.

#### Parameters

- `string sessionKey`

#### Returns

- `array`
  - `struct channelInfo`
    - `int id`
    - `string label`
    - `string name`
    - `string provider_name`
    - `int packages`

- `int systems`
- `string arch_name`

## 7.3. METHOD: LISTPOPULARCHANNELS

List the most popular software channels. Channels that have at least the number of systems subscribed as specified by the popularity count will be returned.

### Parameters

- `string sessionKey`
- `int popularityCount`

### Returns

- `array`
  - `struct channelInfo`
    - `int id`
    - `string label`
    - `string name`
    - `string provider_name`
    - `int packages`
    - `int systems`
    - `string arch_name`

## 7.4. METHOD: LISTREDHATCHANNELS

List all Red Hat software channels that the user's organization is entitled to.

Deprecated - being replaced by `listVendorChannels(String sessionKey)`

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct channelInfo`
    - `int id`
    - `string label`
    - `string name`



- `string provider_name`
- `int packages`
- `int systems`
- `string arch_name`

## 7.5. METHOD: LISTRETIREDCHANNELS

List all retired software channels. These are channels that the user's organization is entitled to, but are no longer supported because they have reached their 'end-of-life' date.

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct channelInfo`
    - `int id`
    - `string label`
    - `string name`
    - `string provider_name`
    - `int packages`
    - `int systems`
    - `string arch_name`

## 7.6. METHOD: LISTSHAREDCHANNELS

List all software channels that may be shared by the user's organization.

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct channelInfo`
    - `int id`
    - `string label`
    - `string name`

- `string provider_name`
- `int packages`
- `int systems`
- `string arch_name`

## 7.7. METHOD: LISTSOFTWARECHANNELS

List all visible software channels.

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct channel`
    - `string label`
    - `string name`
    - `string parent_label`
    - `string end_of_life`
    - `string arch`

## 7.8. METHOD: LISTVENDORCHANNELS

Lists all the vendor software channels that the user's organization is entitled to.

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct channelInfo`
    - `int id`
    - `string label`
    - `string name`
    - `string provider_name`
    - `int packages`

- `int systems`
- `string arch_name`

## CHAPTER 8. NAMESPACE: CHANNEL.ACCESS

Provides methods to retrieve and alter channel access restrictions.

### 8.1. METHOD: DISABLEUSERRESTRICTIONS

Disable user restrictions for the given channel. If disabled, all users within the organization may subscribe to the channel.

#### Parameters

- string `sessionKey`
- string `channelLabel` - label of the channel

#### Returns

- int - 1 on success, exception thrown otherwise.

### 8.2. METHOD: ENABLEUSERRESTRICTIONS

Enable user restrictions for the given channel. If enabled, only selected users within the organization may subscribe to the channel.

#### Parameters

- string `sessionKey`
- string `channelLabel` - label of the channel

#### Returns

- int - 1 on success, exception thrown otherwise.

### 8.3. METHOD: GETORGSCHARING

Get organization sharing access control.

#### Parameters

- string `sessionKey`
- string `channelLabel` - label of the channel

#### Returns

- string - The access value (one of the following: 'public', 'private', or 'protected').

### 8.4. METHOD: SETORGSCHARING

Set organization sharing access control.

#### Parameters

- `string sessionKey`
- `string channelLabel` - label of the channel
- `string access` - Access (one of the following: 'public', 'private', or 'protected')

## Returns

- `int` - 1 on success, exception thrown otherwise.

## CHAPTER 9. NAMESPACE: CHANNEL.ORG

Provides methods to retrieve and alter organization trust relationships for a channel.

### 9.1. METHOD: DISABLEACCESS

Disable access to the channel for the given organization.

#### Parameters

- string `sessionKey`
- string `channelLabel` - label of the channel
- int `orgId` - id of org being removed access

#### Returns

- int - 1 on success, exception thrown otherwise.

### 9.2. METHOD: ENABLEACCESS

Enable access to the channel for the given organization.

#### Parameters

- string `sessionKey`
- string `channelLabel` - label of the channel
- int `orgId` - id of org being granted access

#### Returns

- int - 1 on success, exception thrown otherwise.

### 9.3. METHOD: LIST

List the organizations associated with the given channel that may be trusted.

#### Parameters

- string `sessionKey`
- string `channelLabel` - label of the channel

#### Returns

- array
  - struct `org`
    - int `org_id`
    - string `org_name`

- `boolean access_enabled`

## CHAPTER 10. NAMESPACE: CHANNEL.SOFTWARE

Provides methods to access and modify many aspects of a channel.

### 10.1. METHOD: ADDPACKAGES

Adds a given list of packages to the given channel.

#### Parameters

- `string sessionKey`
- `string channelLabel` - target channel.
- `array`
  - `int packageId` - id of a package to add to the channel.

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 10.2. METHOD: ADDREPOFILTER

Adds a filter for a given repo.

#### Parameters

- `string sessionKey`
- `string label` - repository label
- `struct filter_map`
  - `string filter` - string to filter on
  - `string flag` - + for include, - for exclude

#### Returns

- `int sort` order for new filter

### 10.3. METHOD: ASSOCIATEREPO

Associates a repository with a channel

#### Parameters

- `string sessionKey`
- `string channelLabel` - channel label
- `string repoLabel` - repository label

#### Returns



```
struct channel

int id

string name

string label

string arch_name

string arch_label

string summary

string description

string checksum_label

dateTime.iso8601 last_modified

string maintainer_name

string maintainer_email

string maintainer_phone

string support_policy

string gpg_key_url

string gpg_key_id

string gpg_key_fp

dateTime.iso8601 yumrepo_last_sync - (optional)

string end_of_life

string parent_channel_label

string clone_original

array

    • struct contentSources

        ◦ int id

        ◦ string label

        ◦ string sourceUrl

        ◦ string type
```

## 10.4. METHOD: AVAILABLEENTITLEMENTS

Returns the number of available subscriptions for the given channel

### Parameters

- string `sessionKey`
- string `channelLabel` - channel to query

### Returns

- int number of available subscriptions for the given channel

## 10.5. METHOD: CLEARREPOFILTERS

Removes the filters for a repo

### Parameters

- string `sessionKey`
- string `label` - repository label

### Returns

- int - 1 on success, exception thrown otherwise.

## 10.6. METHOD: CLONE

Clone a channel. If `arch_label` is omitted, the arch label of the original channel will be used. If `parent_label` is omitted, the clone will be a base channel.

### Parameters

- string `sessionKey`
- string `original_label`
- struct `channel` details
  - string `name`
  - string `label`
  - string `summary`
  - string `parent_label` - (optional)
  - string `arch_label` - (optional)
  - string `gpg_key_url` - (optional), `gpg_url` might be used as well
  - string `gpg_key_id` - (optional), `gpg_id` might be used as well
  - string `gpg_key_fp` - (optional), `gpg_fingerprint` might be used as well
  - string `description` - (optional)

- boolean **original\_state**

## Returns

- int the cloned channel ID

## 10.7. METHOD: CREATE

Creates a software channel

### Parameters

- string **sessionKey**
- string **label** - label of the new channel
- string **name** - name of the new channel
- string **summary** - summary of the channel
- string **archLabel** - the label of the architecture the channel corresponds to, see `channel.software.listArches` API for complete listing
- string **parentLabel** - label of the parent of this channel, an empty string if it does not have one
- string **checksumType** - checksum type for this channel, used for yum repository metadata generation
  - **sha1** - Offers widest compatibility with clients
  - **sha256** - Offers highest security, but is compatible only with newer clients: Fedora 11 and newer, or Enterprise Linux 6 and newer.
- struct **gpgKey**
  - string **url** - GPG key URL
  - string **id** - GPG key ID
  - string **fingerprint** - GPG key Fingerprint

## Returns

- int - 1 if the creation operation succeeded, 0 otherwise

Available since: 10.9

## 10.8. METHOD: CREATE

Creates a software channel

### Parameters

- string **sessionKey**

- string **label** - label of the new channel
- string **name** - name of the new channel
- string **summary** - summary of the channel
- string **archLabel** - the label of the architecture the channel corresponds to, see `channel.software.listArches` API for complete listing
- string **parentLabel** - label of the parent of this channel, an empty string if it does not have one
- string **checksumType** - checksum type for this channel, used for yum repository metadata generation
  - **sha1** - Offers widest compatibility with clients
  - **sha256** - Offers highest security, but is compatible only with newer clients: Fedora 11 and newer, or Enterprise Linux 6 and newer.

## Returns

- int - 1 if the creation operation succeeded, 0 otherwise

Available since: 10.9

## 10.9. METHOD: CREATE

Creates a software channel

### Parameters

- string **sessionKey**
- string **label** - label of the new channel
- string **name** - name of the new channel
- string **summary** - summary of the channel
- string **archLabel** - the label of the architecture the channel corresponds to, see `channel.software.listArches` API for complete listing
- string **parentLabel** - label of the parent of this channel, an empty string if it does not have one

## Returns

- int - 1 if the creation operation succeeded, 0 otherwise

## 10.10. METHOD: CREATEREPO

Creates a repository

### Parameters

- string `sessionKey`
- string `label` - repository label
- string `type` - repository type (only YUM is supported)
- string `url` - repository url

## Returns

- struct `channel`
  - int `id`
  - string `label`
  - string `sourceUrl`
  - string `type`

## 10.11. METHOD: DELETE

Deletes a custom software channel

### Parameters

- string `sessionKey`
- string `channelLabel` - channel to delete

### Returns

- int - 1 on success, exception thrown otherwise.

## 10.12. METHOD: DISASSOCIATEREPO

Disassociates a repository from a channel

### Parameters

- string `sessionKey`
- string `channelLabel` - channel label
- string `repoLabel` - repository label

### Returns

struct `channel`

int `id`

string `name`

string `label`

string `arch_name`

`string arch_label`

`string summary`

`string description`

`string checksum_label`

`dateTime.iso8601 last_modified`

`string maintainer_name`

`string maintainer_email`

`string maintainer_phone`

`string support_policy`

`string gpg_key_url`

`string gpg_key_id`

`string gpg_key_fp`

`dateTime.iso8601 yumrepo_last_sync` - (optional)

`string end_of_life`

`string parent_channel_label`

`string clone_original`

`array`

- `struct contentSources`
  - `int id`
  - `string label`
  - `string sourceUrl`
  - `string type`

## 10.13. METHOD: GETCHANNELLASTBUILDID

Returns the last build date of the `repomd.xml` file for the given channel as a localised string.

### Parameters

- `string sessionKey`
- `int id` - id of channel wanted

### Returns

- the last build date of the repomd.xml file as a localised string

## 10.14. METHOD: GETDETAILS

Returns details of the given channel as a map

### Parameters

- string `sessionKey`
- string `channelLabel` - channel to query

### Returns

struct `channel`

int `id`

string `name`

string `label`

string `arch_name`

string `arch_label`

string `summary`

string `description`

string `checksum_label`

dateTime.iso8601 `last_modified`

string `maintainer_name`

string `maintainer_email`

string `maintainer_phone`

string `support_policy`

string `gpg_key_url`

string `gpg_key_id`

string `gpg_key_fp`

dateTime.iso8601 `yumrepo_last_sync` - (optional)

string `end_of_life`

string `parent_channel_label`

string `clone_original`

array

- `struct contentSources`
  - `int id`
  - `string label`
  - `string sourceUrl`
  - `string type`

## 10.15. METHOD: GETDETAILS

Returns details of the given channel as a map

### Parameters

- `string sessionKey`
- `int id` - channel to query

### Returns

`struct channel`

`int id`

`string name`

`string label`

`string arch_name`

`string arch_label`

`string summary`

`string description`

`string checksum_label`

`dateTime.iso8601 last_modified`

`string maintainer_name`

`string maintainer_email`

`string maintainer_phone`

`string support_policy`

`string gpg_key_url`

`string gpg_key_id`

`string gpg_key_fp`

`dateTime.iso8601 yumrepo_last_sync` - (optional)



string `end_of_life`

string `parent_channel_label`

string `clone_original`

array

- struct `contentSources`
  - int `id`
  - string `label`
  - string `sourceUrl`
  - string `type`

## 10.16. METHOD: GETREPODETAILS

Returns details of the given repository

### Parameters

- string `sessionKey`
- string `repoLabel` - repo to query

### Returns

- struct `channel`
  - int `id`
  - string `label`
  - string `sourceUrl`
  - string `type`

## 10.17. METHOD: GETREPODETAILS

Returns details of the given repo

### Parameters

- string `sessionKey`
- string `repoLabel` - repo to query

### Returns

- struct `channel`
  - int `id`

- string `label`
- string `sourceUrl`
- string `type`

## 10.18. METHOD: GETREPOSYNCCRONEXPRESSION

Returns repo synchronization cron expression

### Parameters

- string `sessionKey`
- string `channelLabel` - channel label

### Returns

- string quartz expression

## 10.19. METHOD: ISGLOBALLYSUBSCRIBABLE

Returns whether the channel is subscribable by any user in the organization

### Parameters

- string `sessionKey`
- string `channelLabel` - channel to query

### Returns

- int - 1 if true, 0 otherwise

## 10.20. METHOD: ISUSERMANAGEABLE

Returns whether the channel may be managed by the given user.

### Parameters

- string `sessionKey`
- string `channelLabel` - label of the channel
- string `login` - login of the target user

### Returns

- int - 1 if manageable, 0 if not

## 10.21. METHOD: ISUSERSUBSCRIBABLE

Returns whether the channel may be subscribed to by the given user.

## Parameters

- string `sessionKey`
- string `channelLabel` - label of the channel
- string `login` - login of the target user

## Returns

- int - 1 if subscribable, 0 if not

## 10.22. METHOD: LISTALLPACKAGES

Lists all packages in the channel, regardless of package version, between the given dates.

## Parameters

- string `sessionKey`
- string `channelLabel` - channel to query
- `dateTime.iso8601` `startDate`
- `dateTime.iso8601` `endDate`

## Returns

- array
  - struct `package`
    - string `name`
    - string `version`
    - string `release`
    - string `epoch`
    - string `checksum`
    - string `checksum_type`
    - int `id`
    - string `arch_label`
    - string `last_modified_date`
    - string `last_modified` - (Deprecated)

## 10.23. METHOD: LISTALLPACKAGES

Lists all packages in the channel, regardless of version whose last modified date is greater than given date.

### Parameters

- string `sessionKey`
- string `channelLabel` - channel to query
- dateTime.iso8601 `startDate`

### Returns

- array
  - struct `package`
    - string `name`
    - string `version`
    - string `release`
    - string `epoch`
    - string `checksum`
    - string `checksum_type`
    - int `id`
    - string `arch_label`
    - string `last_modified_date`
    - string `last_modified` - (Deprecated)

## 10.24. METHOD: LISTALLPACKAGES

Lists all packages in the channel, regardless of the package version

### Parameters

- string `sessionKey`
- string `channelLabel` - channel to query

### Returns

- array
  - struct `package`
    - string `name`
    - string `version`

- `string release`
- `string epoch`
- `string checksum`
- `string checksum_type`
- `int id`
- `string arch_label`
- `string last_modified_date`
- `string last_modified` - (Deprecated)

## 10.25. METHOD: LISTALLPACKAGES

Lists all packages in the channel, regardless of package version, between the given dates. Example Date: '2008-08-20 08:00:00'

Deprecated - being replaced by `listAllPackages(string sessionKey, string channelLabel, dateTime.iso8601 startDate, dateTime.iso8601 endDate)`

### Parameters

- `string sessionKey`
- `string channelLabel` - channel to query
- `string startDate`
- `string endDate`

### Returns

- `array`
  - `struct package`
    - `string name`
    - `string version`
    - `string release`
    - `string epoch`
    - `string checksum`
    - `string checksum_type`
    - `int id`
    - `string arch_label`

- `string last_modified_date`
- `string last_modified` - (Deprecated)

## 10.26. METHOD: LISTALLPACKAGES

Lists all packages in the channel, regardless of version whose last modified date is greater than given date. Example Date: '2008-08-20 08:00:00'

Deprecated - being replaced by `listAllPackages(string sessionKey, string channelLabel, dateTime.iso8601 startDate)`

### Parameters

- `string sessionKey`
- `string channelLabel` - channel to query
- `string startDate`

### Returns

- `array`
  - `struct package`
    - `string name`
    - `string version`
    - `string release`
    - `string epoch`
    - `string checksum`
    - `string checksum_type`
    - `int id`
    - `string arch_label`
    - `string last_modified_date`
    - `string last_modified` - (Deprecated)

## 10.27. METHOD: LISTALLPACKAGESBYDATE

Lists all packages in the channel, regardless of the package version, between the given dates. Example Date: '2008-08-20 08:00:00'

Deprecated - being replaced by `listAllPackages(string sessionKey, string channelLabel, dateTime.iso8601 startDate, dateTime.iso8601 endDate)`

### Parameters

- `string sessionKey`
- `string channelLabel` - channel to query
- `string startDate`
- `string endDate`

## Returns

- `array`
  - `struct package`
    - `string name`
    - `string version`
    - `string release`
    - `string epoch`
    - `string id`
    - `string arch_label`
    - `string last_modified`

## 10.28. METHOD: LISTALLPACKAGESBYDATE

Lists all packages in the channel, regardless of the package version, whose last modified date is greater than given date. Example Date: '2008-08-20 08:00:00'

Deprecated - being replaced by `listAllPackages(string sessionKey, string channelLabel, dateTime.iso8601 startDate)`

## Parameters

- `string sessionKey`
- `string channelLabel` - channel to query
- `string startDate`

## Returns

- `array`
  - `struct package`
    - `string name`
    - `string version`
    - `string release`

- `string epoch`
- `string id`
- `string arch_label`
- `string last_modified`

## 10.29. METHOD: LISTALLPACKAGESBYDATE

Lists all packages in the channel, regardless of the package version

Deprecated - being replaced by `listAllPackages(string sessionKey, string channelLabel)`

### Parameters

- `string sessionKey`
- `string channelLabel` - channel to query

### Returns

- `array`
  - `struct package`
    - `string name`
    - `string version`
    - `string release`
    - `string epoch`
    - `string id`
    - `string arch_label`
    - `string last_modified`

## 10.30. METHOD: LISTARCHES

Lists the potential software channel architectures that can be created

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct channel arch`
    - `string name`



- `string label`

## 10.31. METHOD: LISTCHANNELREPOS

Lists associated repos with the given channel

### Parameters

- `string sessionKey`
- `string channelLabel` - channel label

### Returns

- `array`
  - `struct channel`
    - `int id`
    - `string label`
    - `string sourceUrl`
    - `string type`

## 10.32. METHOD: LISTCHILDREN

List the children of a channel

### Parameters

- `string sessionKey`
- `string channelLabel` - the label of the channel

### Returns

`array`

`struct channel`

`int id`

`string name`

`string label`

`string arch_name`

`string arch_label`

`string summary`

`string description`

`string checksum_label`

`dateTime.iso8601 last_modified`

`string maintainer_name`

`string maintainer_email`

`string maintainer_phone`

`string support_policy`

`string gpg_key_url`

`string gpg_key_id`

`string gpg_key_fp`

`dateTime.iso8601 yumrepo_last_sync` - (optional)

`string end_of_life`

`string parent_channel_label`

`string clone_original`

`array`

- `struct contentSources`
  - `int id`
  - `string label`
  - `string sourceUrl`
  - `string type`

## 10.33. METHOD: LISTERRATA

List the errata applicable to a channel after given startDate

### Parameters

- `string sessionKey`
- `string channelLabel` - channel to query
- `dateTime.iso8601 startDate`

### Returns

- `array`
  - `struct errata`

- `int id` - Errata ID.
- `string date` - Date erratum was created.
- `string update_date` - Date erratum was updated.
- `string advisory_synopsis` - Summary of the erratum.
- `string advisory_type` - Type label such as Security, Bug Fix
- `string advisory_name` - Name such as RHSA, etc

## 10.34. METHOD: LISTERRATA

List the errata applicable to a channel between `startDate` and `endDate`.

### Parameters

- `string sessionKey`
- `string channelLabel` - channel to query
- `dateTime.iso8601 startDate`
- `dateTime.iso8601 endDate`

### Returns

- `array`
  - `struct errata`
    - `int id` - Errata ID.
    - `string date` - Date erratum was created.
    - `string update_date` - Date erratum was updated.
    - `string advisory_synopsis` - Summary of the erratum.
    - `string advisory_type` - Type label such as Security, Bug Fix
    - `string advisory_name` - Name such as RHSA, etc

## 10.35. METHOD: LISTERRATA

List the errata applicable to a channel

### Parameters

- `string sessionKey`
- `string channelLabel` - channel to query

### Returns

- **array**
  - **struct errata**
    - **int id** - Errata Id
    - **string advisory\_synopsis** - Summary of the erratum.
    - **string advisory\_type** - Type label such as Security, Bug Fix
    - **string advisory\_name** - Name such as RHSA, etc
    - **string advisory** - name of the advisory (Deprecated)
    - **string issue\_date** - date format follows YYYY-MM-DD HH24:MI:SS (Deprecated)
    - **string update\_date** - date format follows YYYY-MM-DD HH24:MI:SS (Deprecated)
    - **string synopsis** (Deprecated)"
    - **string last\_modified\_date** - date format follows YYYY-MM-DD HH24:MI:SS (Deprecated)

## 10.36. METHOD: LISTERRATA

List the errata applicable to a channel after given startDate

Deprecated - being replaced by `listErrata(string sessionKey, string channelLabel, dateTime.iso8601 startDate)`

### Parameters

- **string sessionKey**
- **string channelLabel** - channel to query
- **string startDate**

### Returns

- **array**
  - **struct errata**
    - **string advisory** - name of the advisory
    - **string issue\_date** - date format follows YYYY-MM-DD HH24:MI:SS
    - **string update\_date** - date format follows YYYY-MM-DD HH24:MI:SS
    - **string synopsis**
    - **string advisory\_type**
    - **string last\_modified\_date** - date format follows YYYY-MM-DD HH24:MI:SS

## 10.37. METHOD: LISTERRATA

List the errata applicable to a channel between startDate and endDate.

Deprecated - being replaced by listErrata(string sessionKey, string channelLabel, dateTime.iso8601 startDate, dateTime.iso8601)

### Parameters

- string sessionKey
- string channelLabel - channel to query
- string startDate
- string endDate

### Returns

- array
  - struct errata
    - string advisory - name of the advisory
    - string issue\_date - date format follows YYYY-MM-DD HH24:MI:SS
    - string update\_date - date format follows YYYY-MM-DD HH24:MI:SS
    - string synopsis
    - string advisory\_type
    - string last\_modified\_date - date format follows YYYY-MM-DD HH24:MI:SS

## 10.38. METHOD: LISTERRATABYTYPE

List the errata of a specific type that are applicable to a channel

### Parameters

- string sessionKey
- string channelLabel - channel to query
- string advisoryType - type of advisory (one of of the following: 'Security Advisory', 'Product Enhancement Advisory', 'Bug Fix Advisory')

### Returns

- array
  - struct errata
    - string advisory - name of the advisory

- string `issue_date` - date format follows YYYY-MM-DD HH24:MI:SS
- string `update_date` - date format follows YYYY-MM-DD HH24:MI:SS
- string `synopsis`
- string `advisory_type`
- string `last_modified_date` - date format follows YYYY-MM-DD HH24:MI:SS

## 10.39. METHOD: LISTERRATANEEEDINGSYNC

If you have satellite-synced a new channel then Red Hat Errata will have been updated with the packages that are in the newly synced channel. A cloned erratum will not have been automatically updated however. If you cloned a channel that includes those cloned errata and should include the new packages, they will not be included when they should. This method lists the errata that will be updated if you run the `syncErrata` method.

### Parameters

- string `sessionKey`
- string `channelLabel` - channel to update

### Returns

- array
  - struct `errata`
    - int `id` - Errata ID.
    - string `date` - Date erratum was created.
    - string `update_date` - Date erratum was updated.
    - string `advisory_synopsis` - Summary of the erratum.
    - string `advisory_type` - Type label such as Security, Bug Fix
    - string `advisory_name` - Name such as RHSA, etc

## 10.40. METHOD: LISTLATESTPACKAGES

Lists the packages with the latest version (including release and epoch) for the given channel

### Parameters

- string `sessionKey`
- string `channelLabel` - channel to query

### Returns

- array

- **struct package**
  - **string name**
  - **string version**
  - **string release**
  - **string epoch**
  - **int id**
  - **string arch\_label**

## 10.41. METHOD: LISTPACKAGESWITHOUTCHANNEL

Lists all packages that are not associated with a channel. Typically these are custom packages.

### Parameters

- **string sessionKey**

### Returns

- **array**
  - **struct package**
    - **string name**
    - **string version**
    - **string release**
    - **string epoch**
    - **int id**
    - **string arch\_label**
    - **string path** - The path on that file system that the package resides
    - **string provider** - The provider of the package, determined by the gpg key it was signed with.
    - **dateTime.iso8601 last\_modified**

## 10.42. METHOD: LISTREPOFILTERS

Lists the filters for a repo

### Parameters

- **string sessionKey**

- string **label** - repository label

## Returns

- array
  - struct **filter**
    - int **sortOrder**
    - string **filter**
    - string **flag**

## 10.43. METHOD: LISTSUBSCRIBEDSYSTEMS

Returns list of subscribed systems for the given channel label

### Parameters

- string **sessionKey**
- string **channelLabel** - channel to query

## Returns

- array
  - struct **system**
    - int **id**
    - string **name**

## 10.44. METHOD: LISTSYSTEMCHANNELS

Returns a list of channels that a system is subscribed to for the given system id

### Parameters

- string **sessionKey**
- int **serverId**

## Returns

- array
  - struct **channel**
    - string **id**
    - string **label**
    - string **name**



## 10.45. METHOD: LISTUSERREPOS

Returns a list of ContentSource (repos) that the user can see

### Parameters

- string `sessionKey`

### Returns

- array
  - struct `map`
    - long `id` - ID of the repo
    - string `label` - label of the repo
    - string `sourceUrl` - URL of the repo

## 10.46. METHOD: MERGEERRATA

Prints a list of merged errata from one channel to another. Use `channel.errata.clone` or `channel.errata.cloneASync` to perform the merge.

### Parameters

- string `sessionKey`
- string `mergeFromLabel` - the label of the channel to pull errata from
- string `mergeToLabel` - the label to push the errata into

### Returns

- array
  - struct `errata`
    - int `id` - Errata Id
    - string `date` - Date erratum was created.
    - string `advisory_type` - Type of the advisory.
    - string `advisory_name` - Name of the advisory.
    - string `advisory_synopsis` - Summary of the erratum.

## 10.47. METHOD: MERGEERRATA

Prints a list of merged errata from one channel to another based upon a given start/end date. Use `channel.errata.clone` or `channel.errata.cloneASync` to perform the merge.

### Parameters

- `string sessionKey`
- `string mergeFromLabel` - the label of the channel to pull errata from
- `string mergeToLabel` - the label to push the errata into
- `string startDate`
- `string endDate`

## Returns

- array
  - `struct errata`
    - `int id` - Errata Id
    - `string date` - Date erratum was created.
    - `string advisory_type` - Type of the advisory.
    - `string advisory_name` - Name of the advisory.
    - `string advisory_synopsis` - Summary of the erratum.

## 10.48. METHOD: MERGEERRATA

Prints a list of merged errata from one channel to another. Use `channel.errata.clone` or `channel.errata.cloneASync` to perform the merge.

## Parameters

- `string sessionKey`
- `string mergeFromLabel` - the label of the channel to pull errata from
- `string mergeToLabel` - the label to push the errata into
- array
  - `string - advisory` - The advisory name of the errata to merge

## Returns

- array
  - `struct errata`
    - `int id` - Errata Id
    - `string date` - Date erratum was created.
    - `string advisory_type` - Type of the advisory.
    - `string advisory_name` - Name of the advisory.

- `string advisory_synopsis` - Summary of the erratum.

## 10.49. METHOD: MERGEPACKAGES

Prints a list of merged packages from one channel to another. Use `channel.software.addPackages` to perform the merge.

### Parameters

- `string sessionKey`
- `string mergeFromLabel` - the label of the channel to pull packages from
- `string mergeToLabel` - the label to push the packages into

### Returns

- `array`
  - `struct package`
    - `string name`
    - `string version`
    - `string release`
    - `string epoch`
    - `int id`
    - `string arch_label`
    - `string path` - The path on that file system that the package resides
    - `string provider` - The provider of the package, determined by the gpg key it was signed with.
    - `dateTime.iso8601 last_modified`

## 10.50. METHOD: REGENERATENEDEDCACHE

Completely clear and regenerate the needed Errata and Package cache for all systems subscribed to the specified channel. This should be used only if you believe your cache is incorrect for all the systems in a given channel.

### Parameters

- `string sessionKey`
- `string channelLabel` - the label of the channel

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 10.51. METHOD: REGENERATENEEEDEDCACHE

Completely clear and regenerate the needed Errata and Package cache for all systems subscribed. You must be a Satellite Admin to perform this action.

### Parameters

- string `sessionKey`

### Returns

- int - 1 on success, exception thrown otherwise.

## 10.52. METHOD: REGENERATEYUMCACHE

Regenerate yum cache for the specified channel.

### Parameters

- string `sessionKey`
- string `channelLabel` - the label of the channel

### Returns

- int - 1 on success, exception thrown otherwise.

## 10.53. METHOD: REMOVEERRATA

Removes a given list of errata from the given channel.

### Parameters

- string `sessionKey`
- string `channelLabel` - target channel.
- array
  - string - `advisoryName` - name of an erratum to remove
- boolean `removePackages` - True to remove packages from the channel

### Returns

- int - 1 on success, exception thrown otherwise.

## 10.54. METHOD: REMOVEPACKAGES

Removes a given list of packages from the given channel.

### Parameters

- string `sessionKey`

- string `channelLabel` - target channel.
- array
  - int `packageId` - id of a package to remove from the channel.

### Returns

- int - 1 on success, exception thrown otherwise.

## 10.55. METHOD: REMOVE\_REPO

Removes a repository

### Parameters

- string `sessionKey`
- long `id` - ID of repo to be removed

### Returns

- int - 1 on success, exception thrown otherwise.

## 10.56. METHOD: REMOVE\_REPO

Removes a repository

### Parameters

- string `sessionKey`
- string `label` - label of repo to be removed

### Returns

- int - 1 on success, exception thrown otherwise.

## 10.57. METHOD: REMOVE\_REPO\_FILTER

Removes a filter for a given repo.

### Parameters

- string `sessionKey`
- string `label` - repository label
- struct `filter_map`
  - string `filter` - string to filter on
  - string `flag` - + for include, - for exclude

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 10.58. METHOD: SETCONTACTDETAILS

Set contact/support information for given channel.

### Parameters

- `string sessionKey`
- `string channelLabel` - label of the channel
- `string maintainerName` - name of the channel maintainer
- `string maintainerEmail` - email of the channel maintainer
- `string maintainerPhone` - phone number of the channel maintainer
- `string supportPolicy` - channel support policy

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 10.59. METHOD: SETDETAILS

Allows to modify channel attributes

### Parameters

- `string sessionKey`
- `int channelId` - channel id
- `struct channel_map`
  - `string checksum_label` - new channel repository checksum label (optional)
  - `string name` - new channel name (optional)
  - `string summary` - new channel summary (optional)
  - `string description` - new channel description (optional)
  - `string maintainer_name` - new channel maintainer name (optional)
  - `string maintainer_email` - new channel email address (optional)
  - `string maintainer_phone` - new channel phone number (optional)
  - `string gpg_key_url` - new channel gpg key url (optional)
  - `string gpg_key_id` - new channel gpg key id (optional)
  - `string gpg_key_fp` - new channel gpg key fingerprint (optional)

## Returns

- int - 1 on success, exception thrown otherwise.

## 10.60. METHOD: SETGLOBALLYSUBSCRIBABLE

Set globally subscribable attribute for given channel.

### Parameters

- string `sessionKey`
- string `channelLabel` - label of the channel
- boolean `subscribable` - true if the channel is to be globally subscribable. False otherwise.

## Returns

- int - 1 on success, exception thrown otherwise.

## 10.61. METHOD: SETREPOFILTERS

Replaces the existing set of filters for a given repo. Filters are ranked by their order in the array.

### Parameters

- string `sessionKey`
- string `label` - repository label
- array
  - struct `filter_map`
    - string `filter` - string to filter on
    - string `flag` - + for include, - for exclude

## Returns

- int - 1 on success, exception thrown otherwise.

## 10.62. METHOD: SETSYSTEMCHANNELS

Change a systems subscribed channels to the list of channels passed in.

Deprecated - being replaced by `system.setBaseChannel(string sessionKey, int serverId, string channelLabel)` and `system.setChildChannels(string sessionKey, int serverId, array[string channelLabel])`

### Parameters

- string `sessionKey`
- int `serverId`

- `array`
  - `string channelLabel` - labels of the channels to subscribe the system to.

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 10.63. METHOD: SETUSERMANAGEABLE

Set the manageable flag for a given channel and user. If value is set to 'true', this method will give the user manage permissions to the channel. Otherwise, that privilege is revoked.

## Parameters

- `string sessionKey`
- `string channelLabel` - label of the channel
- `string login` - login of the target user
- `boolean value` - value of the flag to set

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 10.64. METHOD: SETUSERSUBSCRIBABLE

Set the subscribable flag for a given channel and user. If value is set to 'true', this method will give the user subscribe permissions to the channel. Otherwise, that privilege is revoked.

## Parameters

- `string sessionKey`
- `string channelLabel` - label of the channel
- `string login` - login of the target user
- `boolean value` - value of the flag to set

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 10.65. METHOD: SUBSCRIBESYSTEM

Subscribes a system to a list of channels. If a base channel is included, that is set before setting child channels. When setting child channels the current child channel subscriptions are cleared. To fully unsubscribe the system from all channels, simply provide an empty list of channel labels.

Deprecated - being replaced by `system.setBaseChannel(string sessionKey, int serverId, string channelLabel)` and `system.setChildChannels(string sessionKey, int serverId, array[string channelLabel])`



## Parameters

- `string sessionKey`
- `int serverId`
- `array`
  - `string - label` - channel label to subscribe the system to.

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 10.66. METHOD: SYNCERRATA

If you have satellite-synced a new channel then Red Hat Errata will have been updated with the packages that are in the newly synced channel. A cloned erratum will not have been automatically updated however. If you cloned a channel that includes those cloned errata and should include the new packages, they will not be included when they should. This method updates all the errata in the given cloned channel with packages that have recently been added, and ensures that all the packages you expect are in the channel.

## Parameters

- `string sessionKey`
- `string channelLabel` - channel to update

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 10.67. METHOD: SYNCREPO

Trigger immediate repo synchronization

## Parameters

- `string sessionKey`
- `string channelLabel` - channel label

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 10.68. METHOD: SYNCREPO

Trigger immediate repo synchronization

## Parameters

- `string sessionKey`

- `string channelLabel` - channel label
- `struct params_map`
  - Boolean "sync-kickstars" - Create kickstartable tree - Optional
  - Boolean "no-errata" - Do not sync errata - Optional
  - Boolean "fail" - Terminate upon any error - Optional

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 10.69. METHOD: SYNCREPO

Schedule periodic repo synchronization

### Parameters

- `string sessionKey`
- `string channelLabel` - channel label
- `string cron` expression - if empty all periodic schedules will be disabled

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 10.70. METHOD: SYNCREPO

Schedule periodic repo synchronization

### Parameters

- `string sessionKey`
- `string channelLabel` - channel label
- `string cron` expression - if empty all periodic schedules will be disabled
- `struct params_map`
  - Boolean "sync-kickstars" - Create kickstartable tree - Optional
  - Boolean "no-errata" - Do not sync errata - Optional
  - Boolean "fail" - Terminate upon any error - Optional

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 10.71. METHOD: UPDATEREPO

Updates a ContentSource (repo)

### Parameters

- string `sessionKey`
- int `id` - repository id
- string `label` - new repository label
- string `url` - new repository URL

### Returns

- struct `channel`
  - int `id`
  - string `label`
  - string `sourceUrl`
  - string `type`

## 10.72. METHOD: UPDATEREPOLABEL

Updates repository label

### Parameters

- string `sessionKey`
- int `id` - repository id
- string `label` - new repository label

### Returns

- struct `channel`
  - int `id`
  - string `label`
  - string `sourceUrl`
  - string `type`

## 10.73. METHOD: UPDATEREPOLABEL

Updates repository label

### Parameters

- string `sessionKey`

- string `label` - repository label
- string `newLabel` - new repository label

### Returns

- struct `channel`
  - int `id`
  - string `label`
  - string `sourceUrl`
  - string `type`

## 10.74. METHOD: UPDATEREPOURL

Updates repository source URL

### Parameters

- string `sessionKey`
- int `id` - repository id
- string `url` - new repository url

### Returns

- struct `channel`
  - int `id`
  - string `label`
  - string `sourceUrl`
  - string `type`

## 10.75. METHOD: UPDATEREPOURL

Updates repository source URL

### Parameters

- string `sessionKey`
- string `label` - repository label
- string `url` - new repository url

### Returns

- struct `channel`

- `int id`
- `string label`
- `string sourceUrl`
- `string type`

## CHAPTER 11. NAMESPACE: CONFIGCHANNEL

Provides methods to access and modify many aspects of configuration channels.

### 11.1. METHOD: CHANNELEXISTS

Check for the existence of the config channel provided.

#### Parameters

- `string sessionKey`
- `string channelLabel` - Channel to check for.

#### Returns

- 1 if exists, 0 otherwise.

### 11.2. METHOD: CREATE

Create a new global config channel. Caller must be at least a config admin or an organization admin.

#### Parameters

- `string sessionKey`
- `string channelLabel`
- `string channelName`
- `string channelDescription`

#### Returns

`struct Configuration` Channel information

`int id`

`int orgId`

`string label`

`string name`

`string description`

`struct configChannelType`

`struct Configuration` Channel Type information

- `int id`
- `string label`
- `string name`

- `int priority`

### 11.3. METHOD: CREATEORUPDATEPATH

Create a new file or directory with the given path, or update an existing path.

#### Parameters

- `string sessionKey`
- `string configChannelLabel`
- `string path`
- `boolean isDir` - True if the path is a directory, False if it is a file.
- `struct path info`
  - `string contents` - Contents of the file (text or base64 encoded if binary). (only for non-directories)
  - `boolean contents_enc64` - Identifies base64 encoded content (default: disabled, only for non-directories)
  - `string owner` - Owner of the file/directory.
  - `string group` - Group name of the file/directory.
  - `string permissions` - Octal file/directory permissions (eg: 644)
  - `string selinux_ctx` - SELinux Security context (optional)
  - `string macro-start-delimiter` - Config file macro start delimiter. Use null or empty string to accept the default. (only for non-directories)
  - `string macro-end-delimiter` - Config file macro end delimiter. Use null or empty string to accept the default. (only for non-directories)
  - `int revision` - next revision number, auto increment for null
  - `boolean binary` - mark the binary content, if True, base64 encoded content is expected (only for non-directories)

#### Returns

`struct Configuration` Revision information

`string type`

- `file`
- `directory`
- `symlink`

`string path` - File Path

string **target\_path** - Symbolic link Target File Path. Present for Symbolic links only.

string **channel** - Channel Name

string **contents** - File contents (base64 encoded according to the `contents_enc64` attribute)

boolean **contents\_enc64** - Identifies base64 encoded content

int **revision** - File Revision

dateTime.iso8601 **creation** - Creation Date

dateTime.iso8601 **modified** - Last Modified Date

string **owner** - File Owner. Present for files or directories only.

string **group** - File Group. Present for files or directories only.

int **permissions** - File Permissions (Deprecated). Present for files or directories only.

string **permissions\_mode** - File Permissions. Present for files or directories only.

string **selinux\_ctx** - SELinux Context (optional).

boolean **binary** - true/false , Present for files only.

string **md5** - File's md5 signature. Present for files only.

string **macro-start-delimiter** - Macro start delimiter for a config file. Present for text files only.

string **macro-end-delimiter** - Macro end delimiter for a config file. Present for text files only.

Available since: 10.2

## 11.4. METHOD: CREATEORUPDATESYMLINK

Create a new symbolic link with the given path, or update an existing path.

### Parameters

- string **sessionKey**
- string **configChannelLabel**
- string **path**
- struct **path info**
  - string **target\_path** - The target path for the symbolic link
  - string **selinux\_ctx** - SELinux Security context (optional)
  - int **revision** - next revision number, skip this field for automatic revision number assignment

### Returns



struct **Configuration** Revision information

string **type**

- file
- directory
- symlink

string **path** - File Path

string **target\_path** - Symbolic link Target File Path. Present for Symbolic links only.

string **channel** - Channel Name

string **contents** - File contents (base64 encoded according to the **contents\_enc64** attribute)

boolean **contents\_enc64** - Identifies base64 encoded content

int **revision** - File Revision

dateTime.iso8601 **creation** - Creation Date

dateTime.iso8601 **modified** - Last Modified Date

string **owner** - File Owner. Present for files or directories only.

string **group** - File Group. Present for files or directories only.

int **permissions** - File Permissions (Deprecated). Present for files or directories only.

string **permissions\_mode** - File Permissions. Present for files or directories only.

string **selinux\_ctx** - SELinux Context (optional).

boolean **binary** - true/false , Present for files only.

string **md5** - File's md5 signature. Present for files only.

string **macro-start-delimiter** - Macro start delimiter for a config file. Present for text files only.

string **macro-end-delimiter** - Macro end delimiter for a config file. Present for text files only.

Available since: 10.2

## 11.5. METHOD: DELETECHANNELS

Delete a list of global config channels. Caller must be a config admin.

### Parameters

- string **sessionKey**
- array

- string - configuration channel labels to delete.

## Returns

- int - 1 on success, exception thrown otherwise.

## 11.6. METHOD: DELETEFILEREVISIONS

Delete specified revisions of a given configuration file

### Parameters

- string `sessionKey`
- string `channelLabel` - Label of config channel to lookup on.
- string `filePath` - Configuration file path.
- array
  - int - List of revisions to delete

## Returns

- int - 1 on success, exception thrown otherwise.

## 11.7. METHOD: DELETEFILES

Remove file paths from a global channel.

### Parameters

- string `sessionKey`
- string `channelLabel` - Channel to remove the files from.
- array
  - string - file paths to remove.

## Returns

- int - 1 on success, exception thrown otherwise.

## 11.8. METHOD: DEPLOYALLSYSTEMS

Schedule an immediate configuration deployment for all systems subscribed to a particular configuration channel.

### Parameters

- string `sessionKey`
- string `channelLabel` - The configuration channel's label.

## Returns

- int - 1 on success, exception thrown otherwise.

## 11.9. METHOD: DEPLOYALLSYSTEMS

Schedule a configuration deployment for all systems subscribed to a particular configuration channel.

### Parameters

- string `sessionKey`
- string `channelLabel` - The configuration channel's label.
- `dateTime.iso8601 date` - The date to schedule the action

## Returns

- int - 1 on success, exception thrown otherwise.

## 11.10. METHOD: DEPLOYALLSYSTEMS

Schedule a configuration deployment of a certain file for all systems subscribed to a particular configuration channel.

### Parameters

- string `sessionKey`
- string `channelLabel` - The configuration channel's label.
- string `filePath` - The configuration file path.

## Returns

- int - 1 on success, exception thrown otherwise.

## 11.11. METHOD: DEPLOYALLSYSTEMS

Schedule a configuration deployment of a certain file for all systems subscribed to a particular configuration channel.

### Parameters

- string `sessionKey`
- string `channelLabel` - The configuration channel's label.
- string `filePath` - The configuration file path.
- `dateTime.iso8601 date` - The date to schedule the action

## Returns

- int - 1 on success, exception thrown otherwise.

## 11.12. METHOD: GETDETAILS

Lookup config channel details.

### Parameters

- string `sessionKey`
- string `channelLabel`

### Returns

struct **Configuration** Channel information

int `id`

int `orgId`

string `label`

string `name`

string `description`

struct **configChannelType**

struct **Configuration** Channel Type information

- int `id`
- string `label`
- string `name`
- int `priority`

## 11.13. METHOD: GETDETAILS

Lookup config channel details.

### Parameters

- string `sessionKey`
- int `channelId`

### Returns

struct **Configuration** Channel information

int `id`

int `orgId`

string `label`

string `name`

string **description**

struct **configChannelType**

struct **Configuration** Channel Type information

- int **id**
- string **label**
- string **name**
- int **priority**

## 11.14. METHOD: GETENCODEDFILEREVISION

Get revision of the specified configuration file and transmit the contents as base64 encoded.

### Parameters

- string **sessionKey**
- string **configChannelLabel** - label of config channel to lookup on
- string **filePath** - config file path to examine
- int **revision** - config file revision to examine

### Returns

struct **Configuration** Revision information

string **type**

- file
- directory
- symlink

string **path** - File Path

string **target\_path** - Symbolic link Target File Path. Present for Symbolic links only.

string **channel** - Channel Name

string **contents** - File contents (base64 encoded according to the **contents\_enc64** attribute)

boolean **contents\_enc64** - Identifies base64 encoded content

int **revision** - File Revision

dateTime.iso8601 **creation** - Creation Date

dateTime.iso8601 **modified** - Last Modified Date

string **owner** - File Owner. Present for files or directories only.

string **group** - File Group. Present for files or directories only.

int **permissions** - File Permissions (Deprecated). Present for files or directories only.

string **permissions\_mode** - File Permissions. Present for files or directories only.

string **selinux\_ctx** - SELinux Context (optional).

boolean **binary** - true/false , Present for files only.

string **md5** - File's md5 signature. Present for files only.

string **macro-start-delimiter** - Macro start delimiter for a config file. Present for text files only.

string **macro-end-delimiter** - Macro end delimiter for a config file. Present for text files only.

## 11.15. METHOD: GETFILEREVISION

Get revision of the specified config file

### Parameters

- string **sessionKey**
- string **configChannelLabel** - label of config channel to lookup on
- string **filePath** - config file path to examine
- int **revision** - config file revision to examine

### Returns

struct **Configuration** Revision information

string **type**

- file
- directory
- symlink

string **path** - File Path

string **target\_path** - Symbolic link Target File Path. Present for Symbolic links only.

string **channel** - Channel Name

string **contents** - File contents (base64 encoded according to the contents\_enc64 attribute)

boolean **contents\_enc64** - Identifies base64 encoded content

int **revision** - File Revision

dateTime.iso8601 **creation** - Creation Date

dateTime.iso8601 **modified** - Last Modified Date

string **owner** - File Owner. Present for files or directories only.

string **group** - File Group. Present for files or directories only.

int **permissions** - File Permissions (Deprecated). Present for files or directories only.

string **permissions\_mode** - File Permissions. Present for files or directories only.

string **selinux\_ctx** - SELinux Context (optional).

boolean **binary** - true/false , Present for files only.

string **md5** - File's md5 signature. Present for files only.

string **macro-start-delimiter** - Macro start delimiter for a config file. Present for text files only.

string **macro-end-delimiter** - Macro end delimiter for a config file. Present for text files only.

## 11.16. METHOD: GETFILEREVISIONS

Get list of revisions for specified config file

### Parameters

- string **sessionKey**
- string **channelLabel** - label of config channel to lookup on
- string **filePath** - config file path to examine

### Returns

array

struct **Configuration** Revision information

string **type**

- file
- directory
- symlink

string **path** - File Path

string **target\_path** - Symbolic link Target File Path. Present for Symbolic links only.

string **channel** - Channel Name

string **contents** - File contents (base64 encoded according to the **contents\_enc64** attribute)

boolean **contents\_enc64** - Identifies base64 encoded content

int **revision** - File Revision

dateTime.iso8601 **creation** - Creation Date

`dateTime.iso8601 modified` - Last Modified Date

`string owner` - File Owner. Present for files or directories only.

`string group` - File Group. Present for files or directories only.

`int permissions` - File Permissions (Deprecated). Present for files or directories only.

`string permissions_mode` - File Permissions. Present for files or directories only.

`string selinux_ctx` - SELinux Context (optional).

`boolean binary` - true/false , Present for files only.

`string md5` - File's md5 signature. Present for files only.

`string macro-start-delimiter` - Macro start delimiter for a config file. Present for text files only.

`string macro-end-delimiter` - Macro end delimiter for a config file. Present for text files only.

## 11.17. METHOD: LISTFILES

Return a list of files in a channel.

### Parameters

- `string sessionKey`
- `string channelLabel` - label of config channel to list files on.

### Returns

array

`struct Configuration` File information

`string type`

- `file`
- `directory`
- `symlink`

`string path` - File Path

`dateTime.iso8601 last_modified` - Last Modified Date

## 11.18. METHOD: LISTGLOBALS

List all the global config channels accessible to the logged-in user.

### Parameters

- `string sessionKey`



## Returns

array

struct **Configuration** Channel information

int **id**

int **orgId**

string **label**

string **name**

string **description**

string **type**

struct **configChannelType**

struct **Configuration** Channel Type information

- int **id**
- string **label**
- string **name**
- int **priority**

## 11.19. METHOD: LISTSUBSCRIBEDSYSTEMS

Return a list of systems subscribed to a configuration channel

### Parameters

- string **sessionKey**
- string **channelLabel** - label of config channel to list subscribed systems.

### Returns

- array
  - struct **system**
    - int **id**
    - string **name**

## 11.20. METHOD: LOOKUPCHANNELINFO

Lists details on a list channels given their channel labels.

### Parameters

- `string sessionKey`
- `array`
  - `string` - configuration channel label

## Returns

`array`

`struct Configuration` Channel information

`int id`

`int orgId`

`string label`

`string name`

`string description`

`struct configChannelType`

`struct Configuration` Channel Type information

- `int id`
- `string label`
- `string name`
- `int priority`

## 11.21. METHOD: LOOKUPFILEINFO

Given a list of paths and a channel, returns details about the latest revisions of the paths.

### Parameters

- `string sessionKey`
- `string channelLabel` - label of config channel to lookup on
- `array`
  - `string` - List of paths to examine.

## Returns

`array`

`struct Configuration` Revision information

`string type`

- `file`

- `directory`
- `symlink`

`string path` - File Path

`string target_path` - Symbolic link Target File Path. Present for Symbolic links only.

`string channel` - Channel Name

`string contents` - File contents (base64 encoded according to the `contents_enc64` attribute)

`boolean contents_enc64` - Identifies base64 encoded content

`int revision` - File Revision

`dateTime.iso8601 creation` - Creation Date

`dateTime.iso8601 modified` - Last Modified Date

`string owner` - File Owner. Present for files or directories only.

`string group` - File Group. Present for files or directories only.

`int permissions` - File Permissions (Deprecated). Present for files or directories only.

`string permissions_mode` - File Permissions. Present for files or directories only.

`string selinux_ctx` - SELinux Context (optional).

`boolean binary` - true/false , Present for files only.

`string md5` - File's md5 signature. Present for files only.

`string macro-start-delimiter` - Macro start delimiter for a config file. Present for text files only.

`string macro-end-delimiter` - Macro end delimiter for a config file. Present for text files only.

Available since: 10.2

## 11.22. METHOD: LOOKUPFILEINFO

Given a path, revision number, and a channel, returns details about the latest revisions of the paths.

### Parameters

- `string sessionKey`
- `string channelLabel` - label of config channel to lookup on
- `string path` - path of file/directory
- `int revsion` - The revision number.

### Returns

`struct Configuration` Revision information

**string type**

- file
- directory
- symlink

**string path** - File Path

**string target\_path** - Symbolic link Target File Path. Present for Symbolic links only.

**string channel** - Channel Name

**string contents** - File contents (base64 encoded according to the contents\_enc64 attribute)

**boolean contents\_enc64** - Identifies base64 encoded content

**int revision** - File Revision

**dateTime.iso8601 creation** - Creation Date

**dateTime.iso8601 modified** - Last Modified Date

**string owner** - File Owner. Present for files or directories only.

**string group** - File Group. Present for files or directories only.

**int permissions** - File Permissions (Deprecated). Present for files or directories only.

**string permissions\_mode** - File Permissions. Present for files or directories only.

**string selinux\_ctx** - SELinux Context (optional).

**boolean binary** - true/false , Present for files only.

**string md5** - File's md5 signature. Present for files only.

**string macro-start-delimiter** - Macro start delimiter for a config file. Present for text files only.

**string macro-end-delimiter** - Macro end delimiter for a config file. Present for text files only.

Available since: 10.12

## 11.23. METHOD: SCHEDULEFILECOMPARISONS

Schedule a comparison of the latest revision of a file against the version deployed on a list of systems.

### Parameters

- **string sessionKey**
- **string channelLabel** - Label of config channel
- **string path** - File path

- `array`
  - `long` - The list of server id that the comparison will be performed on

## Returns

- `int actionId` - The action id of the scheduled action

## 11.24. METHOD: UPDATE

Update a global config channel. Caller must be at least a config admin or an organization admin, or have access to a system containing this config channel.

### Parameters

- `string sessionKey`
- `string channelLabel`
- `string channelName`
- `string description`

### Returns

`struct Configuration` Channel information

`int id`

`int orgId`

`string label`

`string name`

`string description`

`struct configChannelType`

`struct Configuration` Channel Type information

- `int id`
- `string label`
- `string name`
- `int priority`

## CHAPTER 12. NAMESPACE: DISTCHANNEL

Provides methods to access and modify distribution channel information

### 12.1. METHOD: LISTDEFAULTMAPS

Lists the default distribution channel maps

#### Parameters

- `string sessionKey`

#### Returns

- `array`
  - `struct distChannelMap`
    - `string os` - Operating System
    - `string release` - OS Release
    - `string arch_name` - Channel architecture
    - `string channel_label` - Channel label
    - `string org_specific` - 'Y' organization specific, 'N' default

### 12.2. METHOD: LISTMAPSFORORG

Lists distribution channel maps valid for the user's organization

#### Parameters

- `string sessionKey`

#### Returns

- `array`
  - `struct distChannelMap`
    - `string os` - Operating System
    - `string release` - OS Release
    - `string arch_name` - Channel architecture
    - `string channel_label` - Channel label
    - `string org_specific` - 'Y' organization specific, 'N' default

### 12.3. METHOD: LISTMAPSFORORG

Lists distribution channel maps valid for an organization, satellite admin right needed

## Parameters

- string `sessionKey`
- int `orgId`

## Returns

- array
  - struct `distChannelMap`
    - string `os` - Operating System
    - string `release` - OS Release
    - string `arch_name` - Channel architecture
    - string `channel_label` - Channel label
    - string `org_specific` - 'Y' organization specific, 'N' default

## 12.4. METHOD: SETMAPFORORG

Sets, overrides (/removes if channelLabel empty) a distribution channel map within an organization

## Parameters

- string `sessionKey`
- string `os`
- string `release`
- string `archName`
- string `channelLabel`

## Returns

- int - 1 on success, exception thrown otherwise.

## CHAPTER 13. NAMESPACE: ERRATA

Provides methods to access and modify errata.

### 13.1. METHOD: ADDPACKAGES

Add a set of packages to an erratum with the given advisory name. This method will only allow for modification of custom errata created either through the UI or API.

#### Parameters

- string `sessionKey`
- string `advisoryName`
- array
  - int `packageId`

#### Returns

- int - representing the number of packages added, exception otherwise

### 13.2. METHOD: APPLICABLETOCHANNELS

Returns a list of channels applicable to the erratum with the given advisory name.

#### Parameters

- string `sessionKey`
- string `advisoryName`

#### Returns

- array
  - struct `channel`
    - int `channel_id`
    - string `label`
    - string `name`
    - string `parent_channel_label`

### 13.3. METHOD: BUGZILLAFIXES

Get the Bugzilla fixes for an erratum matching the given `advisoryName`. The bugs will be returned in a struct where the bug id is the key. i.e. `208144="errata.bugzillaFixes Method Returns different results than docs say"`

#### Parameters

... ..



- string `sessionKey`
- string `advisoryName`

## Returns

- struct `Bugzilla` info
  - string `bugzilla_id` - actual bug number is the key into the struct
  - string `bug_summary` - summary who's key is the bug id

## 13.4. METHOD: CLONE

Clone a list of errata into the specified channel.

### Parameters

- string `sessionKey`
- string `channel_label`
- array
  - string - advisory - The advisory name of the errata to clone.

### Returns

- array
  - struct `errata`
    - int `id` - Errata Id
    - string `date` - Date erratum was created.
    - string `advisory_type` - Type of the advisory.
    - string `advisory_name` - Name of the advisory.
    - string `advisory_synopsis` - Summary of the erratum.

## 13.5. METHOD: CLONEASORIGINAL

Clones a list of errata into a specified cloned channel according the original erratas.

### Parameters

- string `sessionKey`
- string `channel_label`
- array
  - string - advisory - The advisory name of the errata to clone.

## Returns

- array
  - struct errata
    - int id - Errata Id
    - string date - Date erratum was created.
    - string advisory\_type - Type of the advisory.
    - string advisory\_name - Name of the advisory.
    - string advisory\_synopsis - Summary of the erratum.

## 13.6. METHOD: CLONEASORIGINALASYNC

Asynchronously clones a list of errata into a specified cloned channel according the original erratas

### Parameters

- string sessionKey
- string channel\_label
- array
  - string - advisory - The advisory name of the errata to clone.

### Returns

- int - 1 on success, exception thrown otherwise.

## 13.7. METHOD: CLONEASYNC

Asynchronously clone a list of errata into the specified channel.

### Parameters

- string sessionKey
- string channel\_label
- array
  - string - advisory - The advisory name of the errata to clone.

### Returns

- int - 1 on success, exception thrown otherwise.

## 13.8. METHOD: CREATE

Create a custom errata. If "publish" is set to true, the errata will be published as well

## Parameters

- **string** `sessionKey`
- **struct** `errata` info
  - **string** `synopsis`
  - **string** `advisory_name`
  - **int** `advisory_release`
  - **string** `advisory_type` - Type of advisory (one of the following: 'Security Advisory', 'Product Enhancement Advisory', or 'Bug Fix Advisory')
  - **string** `product`
  - **string** `errataFrom`
  - **string** `topic`
  - **string** `description`
  - **string** `references`
  - **string** `notes`
  - **string** `solution`
- **array**
  - **struct** `bug`
    - **int** `id` - Bug Id
    - **string** `summary`
    - **string** `url`
- **array**
  - **string** - keyword - List of keywords to associate with the errata.
- **array**
  - **int** `packageId`
- **boolean** `publish` - Should the errata be published.
- **array**
  - **string** - `channelLabel` - list of channels the errata should be published too, ignored if `publish` is set to false

## Returns

- **struct** `errata`

- `int id` - Errata Id
- `string date` - Date erratum was created.
- `string advisory_type` - Type of the advisory.
- `string advisory_name` - Name of the advisory.
- `string advisory_synopsis` - Summary of the erratum.

## 13.9. METHOD: DELETE

Delete an erratum. This method will only allow for deletion of custom errata created either through the UI or API.

### Parameters

- `string sessionKey`
- `string advisoryName`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 13.10. METHOD: FINDBYCVE

Lookup the details for errata associated with the given CVE (e.g. CVE-2008-3270)

### Parameters

- `string sessionKey`
- `string cveName`

### Returns

- `array`
  - `struct errata`
    - `int id` - Errata Id
    - `string date` - Date erratum was created.
    - `string advisory_type` - Type of the advisory.
    - `string advisory_name` - Name of the advisory.
    - `string advisory_synopsis` - Summary of the erratum.

## 13.11. METHOD: GETDETAILS

Retrieves the details for the erratum matching the given advisory name.

## Parameters

- `string sessionKey`
- `string advisoryName`

## Returns

- `struct erratum`
  - `int id`
  - `string issue_date`
  - `string update_date`
  - `string last_modified_date` - This date is only included for published erratum and it represents the last time the erratum was modified.
  - `string synopsis`
  - `int release`
  - `string type`
  - `string product`
  - `string errataFrom`
  - `string topic`
  - `string description`
  - `string references`
  - `string notes`
  - `string solution`

## 13.12. METHOD: LISTAFFECTEDSYSTEMS

Return the list of systems affected by the erratum with advisory name.

## Parameters

- `string sessionKey`
- `string advisoryName`

## Returns

- `array`
  - `struct system`
    - `int id`

- `string name`
- `dateTime.iso8601 last_checkin` - Last time server successfully checked in

### 13.13. METHOD: LISTBYDATE

List errata that have been applied to a particular channel by date.

Deprecated - being replaced by `channel.software.listErrata(User LoggedInUser, string channelLabel)`

#### Parameters

- `string sessionKey`
- `string channelLabel`

#### Returns

- `array`
  - `struct errata`
    - `int id` - Errata Id
    - `string date` - Date erratum was created.
    - `string advisory_type` - Type of the advisory.
    - `string advisory_name` - Name of the advisory.
    - `string advisory_synopsis` - Summary of the erratum.

### 13.14. METHOD: LISTCVES

Returns a list of

#### Parameters

- `string sessionKey`
- `string advisoryName`

#### Returns

- `array`
  - `string` - cveName

### 13.15. METHOD: LISTKEYWORDS

Get the keywords associated with an erratum matching the given advisory name.

#### Parameters

- `string sessionKey`

- `string advisoryName`

## Returns

- `array`
  - `string` - Keyword associated with erratum.

## 13.16. METHOD: LISTPACKAGES

Returns a list of the packages affected by the erratum with the given advisory name.

## Parameters

- `string sessionKey`
- `string advisoryName`

## Returns

- `array`
  - `struct package`
    - `int id`
    - `string name`
    - `string epoch`
    - `string version`
    - `string release`
    - `string arch_label`
    - `array providing_channels`
      - `string` - Channel label providing this package.
    - `string build_host`
    - `string description`
    - `string checksum`
    - `string checksum_type`
    - `string vendor`
    - `string summary`
    - `string cookie`
    - `string license`

- `string path`
- `string file`
- `string build_date`
- `string last_modified_date`
- `string size`
- `string payload_size`

## 13.17. METHOD: LISTUNPUBLISHEDERRATA

Returns a list of unpublished errata

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct erratum`
    - `int id`
    - `int published`
    - `string advisory`
    - `string advisory_name`
    - `string advisory_type`
    - `string synopsis`
    - `dateTime.iso8601 created`
    - `dateTime.iso8601 update_date`

## 13.18. METHOD: PUBLISH

Publish an existing (unpublished) errata to a set of channels.

### Parameters

- `string sessionKey`
- `string advisoryName`
- `array`
  - `string - channelLabel` - list of channel labels to publish to



## Returns

- **struct errata**
  - **int id** - Errata Id
  - **string date** - Date erratum was created.
  - **string advisory\_type** - Type of the advisory.
  - **string advisory\_name** - Name of the advisory.
  - **string advisory\_synopsis** - Summary of the erratum.

## 13.19. METHOD: PUBLISHASORIGINAL

Publishes an existing (unpublished) cloned errata to a set of cloned channels according to its original erratum

### Parameters

- **string sessionKey**
- **string advisoryName**
- **array**
  - **string - channelLabel** - list of channel labels to publish to

## Returns

- **struct errata**
  - **int id** - Errata Id
  - **string date** - Date erratum was created.
  - **string advisory\_type** - Type of the advisory.
  - **string advisory\_name** - Name of the advisory.
  - **string advisory\_synopsis** - Summary of the erratum.

## 13.20. METHOD: REMOVEPACKAGES

Remove a set of packages from an erratum with the given advisory name. This method will only allow for modification of custom errata created either through the UI or API.

### Parameters

- **string sessionKey**
- **string advisoryName**
- **array**

- `int packageId`

## Returns

- `int` - representing the number of packages removed, exception otherwise

## 13.21. METHOD: SETDETAILS

Set erratum details. All arguments are optional and will only be modified if included in the struct. This method will only allow for modification of custom errata created either through the UI or API.

### Parameters

`string sessionKey`

`string advisoryName`

`struct errata details`

`string synopsis`

`string advisory_name`

`int advisory_release`

`string advisory_type` - Type of advisory (one of the following: 'Security Advisory', 'Product Enhancement Advisory', or 'Bug Fix Advisory')

`string product`

`dateTime.iso8601 issue_date`

`dateTime.iso8601 update_date`

`string errataFrom`

`string topic`

`string description`

`string references`

`string notes`

`string solution`

`array bugs` - 'bugs' is the key into the struct

`array`

- `struct bug`
  - `int id` - Bug Id
  - `string summary`
  - `string url`

array **keywords** - 'keywords' is the key into the struct

array

- string - keyword - List of keywords to associate with the errata.

array **CVEs** - 'cves' is the key into the struct

array

- string - cves - List of CVEs to associate with the errata. (valid only for published errata)

## Returns

- int - 1 on success, exception thrown otherwise.

## CHAPTER 14. NAMESPACE: KICKSTART

Provides methods to create kickstart files

### 14.1. METHOD: CLONEPROFILE

Clone a Kickstart Profile

#### Parameters

- `string sessionKey`
- `string ksLabelToClone` - Label of the kickstart profile to clone
- `string newKsLabel` - label of the cloned profile

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 14.2. METHOD: CREATEPROFILE

Import a kickstart profile into RHN.

#### Parameters

- `string sessionKey`
- `string profileLabel` - Label for the new kickstart profile.
- `string virtualizationType` - none, para\_host, qemu, xenfv or xenpv.
- `string kickstartableTreeLabel` - Label of a kickstartable tree to associate the new profile with.
- `string kickstartHost` - Kickstart hostname (of a satellite or proxy) used to construct the default download URL for the new kickstart profile.
- `string rootPassword` - Root password.
- `string updateType` - Should the profile update itself to use the newest tree available? Possible values are: none (default), red\_hat (only use Kickstart Trees synced from Red Hat), or all (includes custom Kickstart Trees).

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 14.3. METHOD: CREATEPROFILE

Import a kickstart profile into RHN.

#### Parameters

- `string sessionKey`

- string **profileLabel** - Label for the new kickstart profile.
- string **virtualizationType** - none, para\_host, qemu, xenfv or xenpv.
- string **kickstartableTreeLabel** - Label of a kickstartable tree to associate the new profile with.
- string **kickstartHost** - Kickstart hostname (of a satellite or proxy) used to construct the default download URL for the new kickstart profile.
- string **rootPassword** - Root password.

## Returns

- int - 1 on success, exception thrown otherwise.

## 14.4. METHOD: CREATEPROFILEWITHCUSTOMURL

Import a kickstart profile into RHN.

### Parameters

- string **sessionKey**
- string **profileLabel** - Label for the new kickstart profile.
- string **virtualizationType** - none, para\_host, qemu, xenfv or xenpv.
- string **kickstartableTreeLabel** - Label of a kickstartable tree to associate the new profile with.
- boolean **downloadUrl** - Download URL, or 'default' to use the kickstart tree's default URL.
- string **rootPassword** - Root password.

## Returns

- int - 1 on success, exception thrown otherwise.

## 14.5. METHOD: CREATEPROFILEWITHCUSTOMURL

Import a kickstart profile into RHN.

### Parameters

- string **sessionKey**
- string **profileLabel** - Label for the new kickstart profile.
- string **virtualizationType** - none, para\_host, qemu, xenfv or xenpv.
- string **kickstartableTreeLabel** - Label of a kickstartable tree to associate the new profile with.
- boolean **downloadUrl** - Download URL, or 'default' to use the kickstart tree's default URL.

- string **rootPassword** - Root password.
- string **updateType** - Should the profile update itself to use the newest tree available? Possible values are: none (default), red\_hat (only use Kickstart Trees synced from Red Hat), or all (includes custom Kickstart Trees).

## Returns

- int - 1 on success, exception thrown otherwise.

## 14.6. METHOD: DELETEPROFILE

Delete a kickstart profile

### Parameters

- string **sessionKey**
- string **ksLabel** - The label of the kickstart profile you want to remove

## Returns

- int - 1 on success, exception thrown otherwise.

## 14.7. METHOD: DISABLEPROFILE

Enable/Disable a Kickstart Profile

### Parameters

- string **sessionKey**
- string **profileLabel** - Label for the kickstart tree you want to en/disable
- string **disabled** - true to disable the profile

## Returns

- int - 1 on success, exception thrown otherwise.

## 14.8. METHOD: FINDKICKSTARTFORIP

Find an associated kickstart for a given ip address.

### Parameters

- string **sessionKey**
- string **ipAddress** - The ip address to search for (i.e. 192.168.0.1)

## Returns

- string - label of the kickstart. Empty string ("" ) if not found.

## 14.9. METHOD: IMPORTFILE

Import a kickstart profile into RHN.

### Parameters

- string `sessionKey`
- string `profileLabel` - Label for the new kickstart profile.
- string `virtualizationType` - none, para\_host, qemu, xenfv or xenpv.
- string `kickstartableTreeLabel` - Label of a kickstartable tree to associate the new profile with.
- string `kickstartFileContents` - Contents of the kickstart file to import.

### Returns

- int - 1 on success, exception thrown otherwise.

## 14.10. METHOD: IMPORTFILE

Import a kickstart profile into RHN.

### Parameters

- string `sessionKey`
- string `profileLabel` - Label for the new kickstart profile.
- string `virtualizationType` - none, para\_host, qemu, xenfv or xenpv.
- string `kickstartableTreeLabel` - Label of a kickstartable tree to associate the new profile with.
- string `kickstartHost` - Kickstart hostname (of a satellite or proxy) used to construct the default download URL for the new kickstart profile. Using this option signifies that this default URL will be used instead of any url/nfs/cdrom/harddrive commands in the kickstart file itself.
- string `kickstartFileContents` - Contents of the kickstart file to import.

### Returns

- int - 1 on success, exception thrown otherwise.

## 14.11. METHOD: IMPORTFILE

Import a kickstart profile into RHN.

### Parameters

- string `sessionKey`
- string `profileLabel` - Label for the new kickstart profile.

- string **virtualizationType** - none, para\_host, qemu, xenv or xenvp.
- string **kickstartableTreeLabel** - Label of a kickstartable tree to associate the new profile with.
- string **kickstartHost** - Kickstart hostname (of a satellite or proxy) used to construct the default download URL for the new kickstart profile. Using this option signifies that this default URL will be used instead of any url/nfs/cdrom/harddrive commands in the kickstart file itself.
- string **kickstartFileContents** - Contents of the kickstart file to import.
- string **updateType** - Should the profile update itself to use the newest tree available? Possible values are: none (default), red\_hat (only use Kickstart Trees synced from Red Hat), or all (includes custom Kickstart Trees).

## Returns

- int - 1 on success, exception thrown otherwise.

## 14.12. METHOD: IMPORTRAWFILE

Import a raw kickstart file into satellite.

### Parameters

- string **sessionKey**
- string **profileLabel** - Label for the new kickstart profile.
- string **virtualizationType** - none, para\_host, qemu, xenv or xenvp.
- string **kickstartableTreeLabel** - Label of a kickstartable tree to associate the new profile with.
- string **kickstartFileContents** - Contents of the kickstart file to import.

## Returns

- int - 1 on success, exception thrown otherwise.

## 14.13. METHOD: IMPORTRAWFILE

Import a raw kickstart file into satellite.

### Parameters

- string **sessionKey**
- string **profileLabel** - Label for the new kickstart profile.
- string **virtualizationType** - none, para\_host, qemu, xenv or xenvp.
- string **kickstartableTreeLabel** - Label of a kickstartable tree to associate the new profile with.
- string **kickstartFileContents** - Contents of the kickstart file to import.



- string **updateType** - Should the profile update itself to use the newest tree available? Possible values are: none (default), red\_hat (only use Kickstart Trees synced from Red Hat), or all (includes custom Kickstart Trees).

### Returns

- int - 1 on success, exception thrown otherwise.

## 14.14. METHOD: ISPROFILEDISABLED

Returns whether a kickstart profile is disabled

### Parameters

- string **sessionKey**
- string **profileLabel** - kickstart profile label

### Returns

- true if profile is disabled

## 14.15. METHOD: LISTALLIPRANGES

List all Ip Ranges and their associated kickstarts available in the user's org.

### Parameters

- string **sessionKey**

### Returns

- array
  - struct **Kickstart Ip Range**
    - string **ksLabel** - The kickstart label associated with the ip range
    - string **max** - The max ip of the range
    - string **min** - The min ip of the range

## 14.16. METHOD: LISTKICKSTARTABLECHANNELS

List kickstartable channels for the logged in user.

### Parameters

- string **sessionKey**

### Returns

array

struct **channel**

`int id`

`string name`

`string label`

`string arch_name`

`string arch_label`

`string summary`

`string description`

`string checksum_label`

`dateTime.iso8601 last_modified`

`string maintainer_name`

`string maintainer_email`

`string maintainer_phone`

`string support_policy`

`string gpg_key_url`

`string gpg_key_id`

`string gpg_key_fp`

`dateTime.iso8601 yumrepo_last_sync` - (optional)

`string end_of_life`

`string parent_channel_label`

`string clone_original`

`array`

- `struct contentSources`
  - `int id`
  - `string label`
  - `string sourceUrl`
  - `string type`

## 14.17. METHOD: LISTKICKSTARTABLETREES

List the available kickstartable trees for the given channel.

Deprecated - being replaced by `kickstart.tree.list(string sessionKey, string channelLabel)`

### Parameters

- `string sessionKey`
- `string channelLabel` - Label of channel to search.

### Returns

- `array`
  - `struct kickstartable tree`
    - `int id`
    - `string label`
    - `string base_path`
    - `int channel_id`

## 14.18. METHOD: LISTKICKSTARTS

Provides a list of kickstart profiles visible to the user's org

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct kickstart`
    - `string label`
    - `string tree_label`
    - `string name`
    - `boolean advanced_mode`
    - `boolean org_default`
    - `boolean active`
    - `string update_type`

## 14.19. METHOD: RENAMEPROFILE

Rename a Kickstart Profile in Satellite

### Parameters

- string `sessionKey`
- string `originalLabel` - Label for the kickstart profile you want to rename
- string `newLabel` - new label to change to

## Returns

- int - 1 on success, exception thrown otherwise.

## CHAPTER 15. NAMESPACE: KICKSTART.FILEPRESERVATION

Provides methods to retrieve and manipulate kickstart file preservation lists.

### 15.1. METHOD: CREATE

Create a new file preservation list.

#### Parameters

- string `session_key`
- string `name` - name of the file list to create
- array
  - string - name - file names to include

#### Returns

- int - 1 on success, exception thrown otherwise.

### 15.2. METHOD: DELETE

Delete a file preservation list.

#### Parameters

- string `session_key`
- string `name` - name of the file list to delete

#### Returns

- int - 1 on success, exception thrown otherwise.

### 15.3. METHOD: GETDETAILS

Returns all of the data associated with the given file preservation list.

#### Parameters

- string `session_key`
- string `name` - name of the file list to retrieve details for

#### Returns

- struct `file` list
  - string `name`
  - array `file_names`
    - string `name`

## 15.4. METHOD: LISTALLFILEPRESERVATIONS

List all file preservation lists for the organization associated with the user logged into the given session

### Parameters

- string `sessionKey`

### Returns

- array
  - struct `file preservation`
    - int `id`
    - string `name`
    - `dateTime.iso8601 created`
    - `dateTime.iso8601 last_modified`

## CHAPTER 16. NAMESPACE: KICKSTART.KEYS

Provides methods to manipulate kickstart keys.

### 16.1. METHOD: CREATE

creates a new key with the given parameters

#### Parameters

- `string session_key`
- `string description`
- `string type` - valid values are GPG or SSL
- `string content`

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 16.2. METHOD: DELETE

deletes the key identified by the given parameters

#### Parameters

- `string session_key`
- `string description`

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 16.3. METHOD: GETDETAILS

returns all of the data associated with the given key

#### Parameters

- `string session_key`
- `string description`

#### Returns

- `struct key`
  - `string description`
  - `string type`
  - `string content`

## 16.4. METHOD: LISTALLKEYS

list all keys for the org associated with the user logged into the given session

### Parameters

- string `sessionKey`

### Returns

- array
  - struct `key`
    - string `description`
    - string `type`

## 16.5. METHOD: UPDATE

Updates type and content of the key identified by the description

### Parameters

- string `session_key`
- string `description`
- string `type` - valid values are GPG or SSL
- string `content`

### Returns

- int - 1 on success, exception thrown otherwise.



## CHAPTER 17. NAMESPACE: KICKSTART.PROFILE.KEYS

Provides methods to access and modify the list of activation keys associated with a kickstart profile.

### 17.1. METHOD: ADDACTIVATIONKEY

Add an activation key association to the kickstart profile

#### Parameters

- `string sessionKey`
- `string ksLabel` - the kickstart profile label
- `string key` - the activation key

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 17.2. METHOD: GETACTIVATIONKEYS

Lookup the activation keys associated with the kickstart profile.

#### Parameters

- `string sessionKey`
- `string ksLabel` - the kickstart profile label

#### Returns

- `array`
  - `struct activation key`
    - `string key`
    - `string description`
    - `int usage_limit`
    - `string base_channel_label`
    - `array child_channel_labels`
      - `string childChannelLabel`
    - `array entitlements`
      - `string entitlementLabel`
    - `array server_group_ids`
      - `string serverGroupId`

- array **package\_names**
  - string **packageName** - (deprecated by packages)
- array **packages**
  - struct **package**
    - string **name** - packageName
    - string **arch** - archLabel - optional
- boolean **universal\_default**
- boolean **disabled**

## 17.3. METHOD: REMOVEACTIVATIONKEY

Remove an activation key association from the kickstart profile

### Parameters

- string **sessionKey**
- string **ksLabel** - the kickstart profile label
- string **key** - the activation key

### Returns

- int - 1 on success, exception thrown otherwise.

## CHAPTER 18. NAMESPACE: KICKSTART.PROFILE.SOFTWARE

Provides methods to access and modify the software list associated with a kickstart profile.

### 18.1. METHOD: APPENDTOSOFTWARELIST

Append the list of software packages to a kickstart profile. Duplicate packages will be ignored.

#### Parameters

- `string sessionKey`
- `string ksLabel1` - The label of a kickstart profile.
- `string[] packageList` - A list of package names to be added to the profile.

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 18.2. METHOD: GETSOFTWAREDETAILS

Gets kickstart profile software details.

#### Parameters

- `string sessionKey`
- `string ksLabel1` - Label of the kickstart profile

#### Returns

- `struct Kickstart` packages info
  - `string noBase` - Install @Base package group
  - `string ignoreMissing` - Ignore missing packages

### 18.3. METHOD: GETSOFTWARELIST

Get a list of a kickstart profile's software packages.

#### Parameters

- `string sessionKey`
- `string ksLabel1` - The label of a kickstart profile.

#### Returns

- `string[]` - Get a list of a kickstart profile's software packages.

### 18.4. METHOD: SETSOFTWAREDETAILS

Sets kickstart profile software details.

### Parameters

- `string sessionKey`
- `string ksLabel` - Label of the kickstart profile
- `struct Kickstart` packages info
  - `string noBase` - Install @Base package group
  - `string ignoreMissing` - Ignore missing packages

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 18.5. METHOD: SETSOFTWARELIST

Set the list of software packages for a kickstart profile.

### Parameters

- `string sessionKey`
- `string ksLabel` - The label of a kickstart profile.
- `string[] packageList` - A list of package names to be set on the profile.

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 18.6. METHOD: SETSOFTWARELIST

Set the list of software packages for a kickstart profile.

### Parameters

- `string sessionKey`
- `string ksLabel` - The label of a kickstart profile.
- `string[] packageList` - A list of package names to be set on the profile.
- `boolean ignoremissing` - Ignore missing packages if true
- `boolean nobase` - Don't install @Base package group if true

### Returns

- `int` - 1 on success, exception thrown otherwise.

## CHAPTER 19. NAMESPACE: KICKSTART.PROFILE.SYSTEM

Provides methods to set various properties of a kickstart profile.

### 19.1. METHOD: ADDFILEPRESERVATIONS

Adds the given list of file preservations to the specified kickstart profile.

#### Parameters

- string `sessionKey`
- string `kickstartLabel`
- array
  - string - `filePreservations`

#### Returns

- int - 1 on success, exception thrown otherwise.

### 19.2. METHOD: ADDKEYS

Adds the given list of keys to the specified kickstart profile.

#### Parameters

- string `sessionKey`
- string `kickstartLabel`
- array
  - string - `keyDescription`

#### Returns

- int - 1 on success, exception thrown otherwise.

### 19.3. METHOD: CHECKCONFIGMANAGEMENT

Check the configuration management status for a kickstart profile.

#### Parameters

- string `sessionKey`
- string `ksLabel` - the kickstart profile label

#### Returns

- boolean `enabled` - true if configuration management is enabled; otherwise, false

## 19.4. METHOD: CHECKREMOTECOMMANDS

Check the remote commands status flag for a kickstart profile.

### Parameters

- string `sessionKey`
- string `ksLabel1` - the kickstart profile label

### Returns

- boolean `enabled` - true if remote commands support is enabled; otherwise, false

## 19.5. METHOD: DISABLECONFIGMANAGEMENT

Disables the configuration management flag in a kickstart profile so that a system created using this profile will be NOT be configuration capable.

### Parameters

- string `sessionKey`
- string `ksLabel1` - the kickstart profile label

### Returns

- int - 1 on success, exception thrown otherwise.

## 19.6. METHOD: DISABLEREMOTECOMMANDS

Disables the remote command flag in a kickstart profile so that a system created using this profile will be capable of running remote commands

### Parameters

- string `sessionKey`
- string `ksLabel1` - the kickstart profile label

### Returns

- int - 1 on success, exception thrown otherwise.

## 19.7. METHOD: ENABLECONFIGMANAGEMENT

Enables the configuration management flag in a kickstart profile so that a system created using this profile will be configuration capable.

### Parameters

- string `sessionKey`
- string `ksLabel1` - the kickstart profile label

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 19.8. METHOD: ENABLEREMOTECOMMANDS

Enables the remote command flag in a kickstart profile so that a system created using this profile will be capable of running remote commands

### Parameters

- `string sessionKey`
- `string ksLabel1` - the kickstart profile label

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 19.9. METHOD: GETLOCALE

Retrieves the locale for a kickstart profile.

### Parameters

- `string sessionKey`
- `string ksLabel1` - the kickstart profile label

### Returns

`struct locale` info

`string locale`

`boolean useUtc`

- `true` - the hardware clock uses UTC
- `false` - the hardware clock does not use UTC

## 19.10. METHOD: GETPARTITIONINGScheme

Get the partitioning scheme for a kickstart profile.

### Parameters

- `string sessionKey`
- `string ksLabel1` - The label of a kickstart profile.

### Returns

- `string[]` - A list of partitioning commands used to setup the partitions, logical volumes and volume groups."

## 19.11. METHOD: GETREGISTRATIONTYPE

returns the registration type of a given kickstart profile. Registration Type can be one of reactivation/deletion/none These types determine the behaviour of the registration when using this profile for reprovisioning.

### Parameters

- string `sessionKey`
- string `kickstartLabel`

### Returns

- string `registrationType`
  - reactivation
  - deletion
  - none

## 19.12. METHOD: GETSELINUX

Retrieves the SELinux enforcing mode property of a kickstart profile.

### Parameters

- string `sessionKey`
- string `ksLabel` - the kickstart profile label

### Returns

- string `enforcingMode`
  - enforcing
  - permissive
  - disabled

## 19.13. METHOD: LISTFILEPRESERVATIONS

Returns the set of all file preservations associated with the given kickstart profile.

### Parameters

- string `sessionKey`
- string `kickstartLabel`

### Returns

- array



- struct `file` list
  - string `name`
  - array `file_names`
    - string `name`

## 19.14. METHOD: LISTKEYS

Returns the set of all keys associated with the given kickstart profile.

### Parameters

- string `sessionKey`
- string `kickstartLabel`

### Returns

- array
  - struct `key`
    - string `description`
    - string `type`
    - string `content`

## 19.15. METHOD: REMOVEFILEPRESERVATIONS

Removes the given list of file preservations from the specified kickstart profile.

### Parameters

- string `sessionKey`
- string `kickstartLabel`
- array
  - string - `filePreservations`

### Returns

- int - 1 on success, exception thrown otherwise.

## 19.16. METHOD: REMOVEKEYS

Removes the given list of keys from the specified kickstart profile.

### Parameters

- string `sessionKey`

- `string kickstartLabel`
- `array`
  - `string` - `keyDescription`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 19.17. METHOD: SETLOCALE

Sets the locale for a kickstart profile.

### Parameters

- `string sessionKey`
- `string ksLabel` - the kickstart profile label
- `string locale` - the locale
- `boolean useUtc`
  - `true` - the hardware clock uses UTC
  - `false` - the hardware clock does not use UTC

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 19.18. METHOD: SETPARTITIONINGScheme

Set the partitioning scheme for a kickstart profile.

### Parameters

- `string sessionKey`
- `string ksLabel` - The label of the kickstart profile to update.
- `string[] scheme` - The partitioning scheme is a list of partitioning command strings used to setup the partitions, volume groups and logical volumes.

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 19.19. METHOD: SETREGISTRATIONTYPE

Sets the registration type of a given kickstart profile. Registration Type can be one of reactivation/deletion/none These types determine the behaviour of the re registration when using this profile.

## Parameters

- string `sessionKey`
- string `kickstartLabel`
- string `registrationType`
  - `reactivation` - to try and generate a reactivation key and use that to register the system when reprovisioning a system.
  - `deletion` - to try and delete the existing system profile and reregister the system being reprovisioned as new
  - `none` - to preserve the status quo and leave the current system as a duplicate on a reprovision.

## Returns

- int - 1 on success, exception thrown otherwise.

## 19.20. METHOD: SETSELINUX

Sets the SELinux enforcing mode property of a kickstart profile so that a system created using this profile will have the appropriate SELinux enforcing mode.

## Parameters

- string `sessionKey`
- string `ksLabel` - the kickstart profile label
- string `enforcingMode` - the selinux enforcing mode
  - `enforcing`
  - `permissive`
  - `disabled`

## Returns

- int - 1 on success, exception thrown otherwise.

## CHAPTER 20. NAMESPACE: KICKSTART.PROFILE

Provides methods to access and modify many aspects of a kickstart profile.

### 20.1. METHOD: ADDIPRANGE

Add an ip range to a kickstart profile.

#### Parameters

- `string sessionKey`
- `string label` - The label of the kickstart
- `string min` - The ip address making up the minimum of the range (i.e. 192.168.0.1)
- `string max` - The ip address making up the maximum of the range (i.e. 192.168.0.254)

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 20.2. METHOD: ADDSCRIPT

Add a pre/post script to a kickstart profile.

#### Parameters

- `string sessionKey`
- `string ksLabel` - The kickstart label to add the script to.
- `string name` - The kickstart script name.
- `string contents` - The full script to add.
- `string interpreter` - The path to the interpreter to use (i.e. /bin/bash). An empty string will use the kickstart default interpreter.
- `string type` - The type of script (either 'pre' or 'post').
- `boolean chroot` - Whether to run the script in the chrooted install location (recommended) or not.

#### Returns

- `int id` - the id of the added script

### 20.3. METHOD: ADDSCRIPT

Add a pre/post script to a kickstart profile.

#### Parameters

- `string sessionKey`

- string **ksLabel** - The kickstart label to add the script to.
- string **name** - The kickstart script name.
- string **contents** - The full script to add.
- string **interpreter** - The path to the interpreter to use (i.e. /bin/bash). An empty string will use the kickstart default interpreter.
- string **type** - The type of script (either 'pre' or 'post').
- boolean **chroot** - Whether to run the script in the chrooted install location (recommended) or not.
- boolean **template** - Enable templating using cobbler.

## Returns

- int **id** - the id of the added script

## 20.4. METHOD: ADDSCRIPT

Add a pre/post script to a kickstart profile.

### Parameters

- string **sessionKey**
- string **ksLabel** - The kickstart label to add the script to.
- string **name** - The kickstart script name.
- string **contents** - The full script to add.
- string **interpreter** - The path to the interpreter to use (i.e. /bin/bash). An empty string will use the kickstart default interpreter.
- string **type** - The type of script (either 'pre' or 'post').
- boolean **chroot** - Whether to run the script in the chrooted install location (recommended) or not.
- boolean **template** - Enable templating using cobbler.
- boolean **erroronfail** - Whether to throw an error if the script fails or not

## Returns

- int **id** - the id of the added script

## 20.5. METHOD: COMPAREACTIVATIONKEYS

Returns a list for each kickstart profile; each list will contain activation keys not present on the other profile.

## Parameters

- string `sessionKey`
- string `kickstartLabel1`
- string `kickstartLabel2`

## Returns

struct `ComparisonInfo`

array `kickstartLabel1` - Actual label of the first kickstart profile is the key into the struct

array

- struct `activationKey`
  - string `key`
  - string `description`
  - int `usage_limit`
  - string `base_channel_label`
  - array `child_channel_labels`
    - string `childChannelLabel`
  - array `entitlements`
    - string `entitlementLabel`
  - array `server_group_ids`
    - string `serverGroupId`
  - array `package_names`
    - string `packageName` - (deprecated by packages)
  - array `packages`
    - struct `package`
      - string `name` - `packageName`
      - string `arch` - `archLabel` - optional
  - boolean `universal_default`
  - boolean `disabled`

array `kickstartLabel2` - Actual label of the second kickstart profile is the key into the struct

array

- **struct activation key**
  - **string key**
  - **string description**
  - **int usage\_limit**
  - **string base\_channel\_label**
  - **array child\_channel\_labels**
    - **string childChannelLabel**
  - **array entitlements**
    - **string entitlementLabel**
  - **array server\_group\_ids**
    - **string serverGroupId**
  - **array package\_names**
    - **string packageName** - (deprecated by packages)
  - **array packages**
    - **struct package**
      - **string name** - packageName
      - **string arch** - archLabel - optional
  - **boolean universal\_default**
  - **boolean disabled**

## 20.6. METHOD: COMPAREADVANCEDOPTIONS

Returns a list for each kickstart profile; each list will contain the properties that differ between the profiles and their values for that specific profile .

### Parameters

- **string sessionKey**
- **string kickstartLabel1**
- **string kickstartLabel2**

### Returns

**struct Comparison Info**

**array kickstartLabel1** - Actual label of the first kickstart profile is the key into the struct

array

- **struct value**
  - string **name**
  - string **value**
  - boolean **enabled**

array **kickstartLabel2** - Actual label of the second kickstart profile is the key into the struct

array

- **struct value**
  - string **name**
  - string **value**
  - boolean **enabled**

## 20.7. METHOD: COMPAREPACKAGES

Returns a list for each kickstart profile; each list will contain package names not present on the other profile.

### Parameters

- string **sessionKey**
- string **kickstartLabel1**
- string **kickstartLabel2**

### Returns

struct **Comparison Info**

array **kickstartLabel1** - Actual label of the first kickstart profile is the key into the struct

array

- string - package name

array **kickstartLabel2** - Actual label of the second kickstart profile is the key into the struct

array

- string - package name

## 20.8. METHOD: DOWNLOADKICKSTART

Download the full contents of a kickstart file.

### Parameters



- string `sessionKey`
- string `ksLabel` - The label of the kickstart to download.
- string `host` - The host to use when referring to the satellite itself (Usually this should be the FQDN of the satellite, but could be the ip address or shortname of it as well).

## Returns

- string - The contents of the kickstart file. Note: if an activation key is not associated with the kickstart file, registration will not occur in the satellite generated %post section. If one is associated, it will be used for registration.

## 20.9. METHOD: DOWNLOADRENDEREDKICKSTART

Downloads the Cobbler-rendered Kickstart file.

### Parameters

- string `sessionKey`
- string `ksLabel` - The label of the kickstart to download.

## Returns

- string - The contents of the kickstart file.

## 20.10. METHOD: GETADVANCEDOPTIONS

Get advanced options for a kickstart profile.

### Parameters

- string `sessionKey`
- string `ksLabel` - Label of kickstart profile to be changed.

## Returns

- array
  - struct `option`
    - string `name`
    - string `arguments`

## 20.11. METHOD: GETAVAILABLEREPOSITORIES

Lists available OS repositories to associate with the provided kickstart profile.

### Parameters

- string `sessionKey`

- string `ksLabel`

## Returns

- array
  - string - repositoryLabel

## 20.12. METHOD: GETCFGPRESERVATION

Get ks.cfg preservation option for a kickstart profile.

### Parameters

- string `sessionKey`
- string `kslabel` - Label of kickstart profile to be changed.

## Returns

- boolean - The value of the option. True means that ks.cfg will be copied to /root, false means that it will not.

## 20.13. METHOD: GETCHILDCHANNELS

Get the child channels for a kickstart profile.

### Parameters

- string `sessionKey`
- string `kslabel` - Label of kickstart profile.

## Returns

- array
  - string - channelLabel

## 20.14. METHOD: GETCUSTOMOPTIONS

Get custom options for a kickstart profile.

### Parameters

- string `sessionKey`
- string `ksLabel`

## Returns

- array
  - struct `option`

- `int id`
- `string arguments`

## 20.15. METHOD: GETKICKSTARTTREE

Get the kickstart tree for a kickstart profile.

### Parameters

- `string sessionKey`
- `string kslabel` - Label of kickstart profile to be changed.

### Returns

- `string kstreeLabel` - Label of the kickstart tree.

## 20.16. METHOD: GETREPOSITORIES

Lists all OS repositories associated with provided kickstart profile.

### Parameters

- `string sessionKey`
- `string ksLabel`

### Returns

- `array`
  - `string` - repositoryLabel

## 20.17. METHOD: GETUPDATETYPE

Get the update type for a kickstart profile.

### Parameters

- `string sessionKey`
- `string kslabel` - Label of kickstart profile.

### Returns

- `string update_type` - Update type for this Kickstart Profile.

## 20.18. METHOD: GETVARIABLES

Returns a list of variables associated with the specified kickstart profile

### Parameters

- `string sessionKey`

- string `ksLabel`

## Returns

- struct `kickstart` variable
  - string `key`
  - string or int "value"

## 20.19. METHOD: GETVIRTUALIZATIONTYPE

For given kickstart profile label returns label of virtualization type it's using

### Parameters

- string `sessionKey`
- string `ksLabel`

### Returns

- string `virtLabel` - Label of virtualization type.

## 20.20. METHOD: LISTIPRANGES

List all ip ranges for a kickstart profile.

### Parameters

- string `sessionKey`
- string `label` - The label of the kickstart

### Returns

- array
  - struct `Kickstart Ip Range`
    - string `ksLabel` - The kickstart label associated with the ip range
    - string `max` - The max ip of the range
    - string `min` - The min ip of the range

## 20.21. METHOD: LISTSCRIPTS

List the pre and post scripts for a kickstart profile in the order they will run during the kickstart.

### Parameters

- string `sessionKey`
- string `ksLabel` - The label of the kickstart

## Returns

- array
  - struct `kickstart script`
    - `int id`
    - `string name`
    - `string contents`
    - `string script_type` - Which type of script ('pre' or 'post').
    - `string interpreter` - The scripting language interpreter to use for this script. An empty string indicates the default kickstart shell.
    - `boolean chroot` - True if the script will be executed within the chroot environment.
    - `boolean erroronfail` - True if the script will throw an error if it fails.
    - `boolean template` - True if templating using cobbler is enabled
    - `boolean beforeRegistration` - True if script will run before the server registers and performs server actions.

## 20.22. METHOD: ORDERSCRIPTS

Change the order that kickstart scripts will run for this kickstart profile. Scripts will run in the order they appear in the array. There are three arrays, one for all pre scripts, one for the post scripts that run before registration and server actions happen, and one for post scripts that run after registration and server actions. All scripts must be included in one of these lists, as appropriate.

## Parameters

- `string sessionKey`
- `string ksLabel` - The label of the kickstart
- array
  - `int` - IDs of the ordered pre scripts
- array
  - `int` - IDs of the ordered post scripts that will run before registration
- array
  - `int` - IDs of the ordered post scripts that will run after registration

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 20.23. METHOD: REMOVEIPRANGE

Remove an ip range from a kickstart profile.

### Parameters

- string `sessionKey`
- string `ksLabel` - The kickstart label of the ip range you want to remove
- string `ip_address` - An Ip Address that falls within the range that you are wanting to remove. The min or max of the range will work.

### Returns

- int - 1 on successful removal, 0 if range wasn't found for the specified kickstart, exception otherwise.

## 20.24. METHOD: REMOVESCRIPT

Remove a script from a kickstart profile.

### Parameters

- string `sessionKey`
- string `ksLabel` - The kickstart from which to remove the script from.
- int `scriptId` - The id of the script to remove.

### Returns

- int - 1 on success, exception thrown otherwise.

## 20.25. METHOD: SETADVANCEDOPTIONS

Set advanced options for a kickstart profile. If 'md5\_crypt\_rootpw' is set to 'True', 'root\_pw' is taken as plaintext and will md5 encrypted on server side, otherwise a hash encoded password (according to the auth option) is expected

### Parameters

- string `sessionKey`
- string `ksLabel`
- array
  - struct `advanced options`
    - string `name` - Name of the advanced option. Valid Option names: autostep, interactive, install, upgrade, text, network, cdrom, harddrive, nfs, url, lang, langsupport keyboard, mouse, device, deviceprobe, zerombr, clearpart, bootloader, timezone, auth, rootpw, selinux, reboot, firewall, xconfig, skipx, key, ignoredisk, autopart, cmdline, firstboot, graphical, iscsi, iscsiname, logging, monitor, multipath, poweroff, halt, services, shutdown, user, vnc, zfc, driverdisk, md5\_crypt\_rootpw
    - string `arguments` - Arguments of the option

## Returns

- int - 1 on success, exception thrown otherwise.

## 20.26. METHOD: SETCFGPRESERVATION

Set ks.cfg preservation option for a kickstart profile.

### Parameters

- string `sessionKey`
- string `kslabel` - Label of kickstart profile to be changed.
- boolean `preserve` - whether or not ks.cfg and all %include fragments will be copied to /root.

## Returns

- int - 1 on success, exception thrown otherwise.

## 20.27. METHOD: SETCHILDCHANNELS

Set the child channels for a kickstart profile.

### Parameters

- string `sessionKey`
- string `kslabel` - Label of kickstart profile to be changed.
- string[] `channelLabels` - List of labels of child channels

## Returns

- int - 1 on success, exception thrown otherwise.

## 20.28. METHOD: SETCUSTOMOPTIONS

Set custom options for a kickstart profile.

### Parameters

- string `sessionKey`
- string `ksLabel`
- string[] `options`

## Returns

- int - 1 on success, exception thrown otherwise.

## 20.29. METHOD: SETKICKSTARTTREE

Set the kickstart tree for a kickstart profile.

## Parameters

- string `sessionKey`
- string `kslabel` - Label of kickstart profile to be changed.
- string `kstreeLabel` - Label of new kickstart tree.

## Returns

- int - 1 on success, exception thrown otherwise.

## 20.30. METHOD: SETLOGGING

Set logging options for a kickstart profile.

### Parameters

- string `sessionKey`
- string `kslabel` - Label of kickstart profile to be changed.
- boolean `pre` - whether or not to log the pre section of a kickstart to `/root/ks-pre.log`
- boolean `post` - whether or not to log the post section of a kickstart to `/root/ks-post.log`

### Returns

- int - 1 on success, exception thrown otherwise.

## 20.31. METHOD: SETREPOSITORIES

\$call.doc

### Parameters

- string `sessionKey`
- string `ksLabel`
- array
  - string - repositoryLabel

### Returns

- int - 1 on success, exception thrown otherwise.

## 20.32. METHOD: SETUPDATETYPE

Set the update type for a kickstart profile.

### Parameters

- string `sessionKey`



- string `ksLabel` - Label of kickstart profile to be changed.
- string `updateType` - The new update type to set. Possible values are 'red\_hat', 'all', and 'none'.

### Returns

- int - 1 on success, exception thrown otherwise.

## 20.33. METHOD: SETVARIABLES

Associates list of kickstart variables with the specified kickstart profile

### Parameters

- string `sessionKey`
- string `ksLabel`
- struct `kickstart` variable
  - string `key`
  - string or int "value"

### Returns

- int - 1 on success, exception thrown otherwise.

## 20.34. METHOD: SETVIRTUALIZATIONTYPE

For given kickstart profile label sets its virtualization type.

### Parameters

- string `sessionKey`
- string `ksLabel`
- string `typeLabel` - One of the following: 'none', 'qemu', 'para\_host', 'xenpv', 'xenfv'

### Returns

- int - 1 on success, exception thrown otherwise.

## CHAPTER 21. NAMESPACE: KICKSTART.SNIPPET

Provides methods to create kickstart files

### 21.1. METHOD: CREATEORUPDATE

Will create a snippet with the given name and contents if it doesn't exist. If it does exist, the existing snippet will be updated.

#### Parameters

- `string sessionKey`
- `string name`
- `string contents`

#### Returns

- `struct snippet`
  - `string name`
  - `string contents`
  - `string fragment` - The string to include in a kickstart file that will generate this snippet.
  - `string file` - The local path to the file containing this snippet.

### 21.2. METHOD: DELETE

Delete the specified snippet. If the snippet is not found, 0 is returned.

#### Parameters

- `string sessionKey`
- `string name`

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 21.3. METHOD: LISTALL

List all cobbler snippets for the logged in user

#### Parameters

- `string sessionKey`

#### Returns

- `array`

- **struct snippet**
  - **string name**
  - **string contents**
  - **string fragment** - The string to include in a kickstart file that will generate this snippet.
  - **string file** - The local path to the file containing this snippet.

## 21.4. METHOD: LISTCUSTOM

List only custom snippets for the logged in user. These snippets are editable.

### Parameters

- **string sessionKey**

### Returns

- **array**
  - **struct snippet**
    - **string name**
    - **string contents**
    - **string fragment** - The string to include in a kickstart file that will generate this snippet.
    - **string file** - The local path to the file containing this snippet.

## 21.5. METHOD: LISTDEFAULT

List only pre-made default snippets for the logged in user. These snippets are not editable.

### Parameters

- **string sessionKey**

### Returns

- **array**
  - **struct snippet**
    - **string name**
    - **string contents**
    - **string fragment** - The string to include in a kickstart file that will generate this snippet.
    - **string file** - The local path to the file containing this snippet.



## CHAPTER 22. NAMESPACE: KICKSTART.TREE

Provides methods to access and modify the kickstart trees.

### 22.1. METHOD: CREATE

Create a Kickstart Tree (Distribution) in Satellite.

#### Parameters

- `string sessionKey`
- `string treeLabel` - The new kickstart tree label.
- `string basePath` - Path to the base or root of the kickstart tree.
- `string channelLabel` - Label of channel to associate with the kickstart tree.
- `string installType` - Label for KickstartInstallType (rhel\_2.1, rhel\_3, rhel\_4, rhel\_5, fedora\_9).

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 22.2. METHOD: DELETE

Delete a Kickstart Tree (Distribution) in Satellite.

#### Parameters

- `string sessionKey`
- `string treeLabel` - Label for the kickstart tree to delete.

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 22.3. METHOD: DELETETREEANDPROFILES

Delete a kickstarttree and any profiles associated with this kickstart tree. WARNING: This will delete all profiles associated with this kickstart tree!

#### Parameters

- `string sessionKey`
- `string treeLabel` - Label for the kickstart tree to delete.

#### Returns

- `int` - 1 on success, exception thrown otherwise.

## 22.4. METHOD: GETDETAILS

The detailed information about a kickstartable tree given the tree name.

### Parameters

- string `sessionKey`
- string `treeLabel` - Label of kickstartable tree to search.

### Returns

struct `kickstartable tree`

int `id`

string `label`

string `abs_path`

int `channel_id`

struct `kickstart install type`

- int `id`
- string `label`
- string `name`

## 22.5. METHOD: LIST

List the available kickstartable trees for the given channel.

### Parameters

- string `sessionKey`
- string `channelLabel` - Label of channel to search.

### Returns

- array
  - struct `kickstartable tree`
    - int `id`
    - string `label`
    - string `base_path`
    - int `channel_id`

## 22.6. METHOD: LISTINSTALLTYPES

List the available kickstartable install types (rhel2,3,4,5 and fedora9+).

### Parameters

- string `sessionKey`

### Returns

- array
  - struct `kickstart` install type
    - int `id`
    - string `label`
    - string `name`

## 22.7. METHOD: RENAME

Rename a Kickstart Tree (Distribution) in Satellite.

### Parameters

- string `sessionKey`
- string `originalLabel` - Label for the kickstart tree to rename.
- string `newLabel` - The kickstart tree's new label.

### Returns

- int - 1 on success, exception thrown otherwise.

## 22.8. METHOD: UPDATE

Edit a Kickstart Tree (Distribution) in Satellite.

### Parameters

- string `sessionKey`
- string `treeLabel` - Label for the kickstart tree.
- string `basePath` - Path to the base or root of the kickstart tree.
- string `channelLabel` - Label of channel to associate with kickstart tree.
- string `installType` - Label for `KickstartInstallType` (rhel\_2.1, rhel\_3, rhel\_4, rhel\_5, fedora\_9).

### Returns

- int - 1 on success, exception thrown otherwise.

## CHAPTER 23. NAMESPACE: ORG

Contains methods to access common organization management functions available from the web interface.

### 23.1. METHOD: CREATE

Create a new organization and associated administrator account.

#### Parameters

- string **sessionKey**
- string **orgName** - Organization name. Must meet same criteria as in the web UI.
- string **adminLogin** - New administrator login name.
- string **adminPassword** - New administrator password.
- string **prefix** - New administrator's prefix. Must match one of the values available in the web UI. (i.e. Dr., Mr., Mrs., Sr., etc.)
- string **firstName** - New administrator's first name.
- string **lastName** - New administrator's first name.
- string **email** - New administrator's e-mail.
- boolean **usePamAuth** - True if PAM authentication should be used for the new administrator account.

#### Returns

- struct **organization** info
  - int **id**
  - string **name**
  - int **active\_users** - Number of active users in the organization.
  - int **systems** - Number of systems in the organization.
  - int **trusts** - Number of trusted organizations.
  - int **system\_groups** - Number of system groups in the organization. (optional)
  - int **activation\_keys** - Number of activation keys in the organization. (optional)
  - int **kickstart\_profiles** - Number of kickstart profiles in the organization. (optional)
  - int **configuration\_channels** - Number of configuration channels in the organization. (optional)



- boolean `staging_content_enabled` - Is staging content enabled in organization. (optional)

## 23.2. METHOD: DELETE

Delete an organization. The default organization (i.e. `orgId=1`) cannot be deleted.

### Parameters

- string `sessionKey`
- int `orgId`

### Returns

- int - 1 on success, exception thrown otherwise.

## 23.3. METHOD: GETCRASHFILESIZELIMIT

Get the organization wide crash file size limit. The limit value must be a non-negative number, zero means no limit.

### Parameters

- string `sessionKey`
- int `orgId`

### Returns

- int - Crash file size limit.

## 23.4. METHOD: GETDETAILS

The detailed information about an organization given the organization ID.

### Parameters

- string `sessionKey`
- int `orgId`

### Returns

- struct `organization info`
  - int `id`
  - string `name`
  - int `active_users` - Number of active users in the organization.
  - int `systems` - Number of systems in the organization.
  - int `trusts` - Number of trusted organizations.

- `int system_groups` - Number of system groups in the organization. (optional)
- `int activation_keys` - Number of activation keys in the organization. (optional)
- `int kickstart_profiles` - Number of kickstart profiles in the organization. (optional)
- `int configuration_channels` - Number of configuration channels in the organization. (optional)
- `boolean staging_content_enabled` - Is staging content enabled in organization. (optional)

## 23.5. METHOD: GETDETAILS

The detailed information about an organization given the organization name.

### Parameters

- `string sessionKey`
- `string name`

### Returns

- `struct organization info`
  - `int id`
  - `string name`
  - `int active_users` - Number of active users in the organization.
  - `int systems` - Number of systems in the organization.
  - `int trusts` - Number of trusted organizations.
  - `int system_groups` - Number of system groups in the organization. (optional)
  - `int activation_keys` - Number of activation keys in the organization. (optional)
  - `int kickstart_profiles` - Number of kickstart profiles in the organization. (optional)
  - `int configuration_channels` - Number of configuration channels in the organization. (optional)
  - `boolean staging_content_enabled` - Is staging content enabled in organization. (optional)

## 23.6. METHOD: GETPOLICYFORSCAPFILEUPLOAD

Get the status of SCAP detailed result file upload settings for the given organization.

### Parameters

- `string sessionKey`

- `int orgId`

## Returns

- `struct scap_upload_info`
  - `boolean enabled` - Aggregation of detailed SCAP results is enabled.
  - `int size_limit` - Limit (in Bytes) for a single SCAP file upload.

## 23.7. METHOD: GETPOLICYFORSCAPRESULTDELETION

Get the status of SCAP result deletion settings for the given organization.

### Parameters

- `string sessionKey`
- `int orgId`

## Returns

- `struct scap_deletion_info`
  - `boolean enabled` - Deletion of SCAP results is enabled
  - `int retention_period` - Period (in days) after which a scan can be deleted (if enabled).

## 23.8. METHOD: ISCRASHREPORTINGENABLED

Get the status of crash reporting settings for the given organization. Returns true if enabled, false otherwise.

### Parameters

- `string sessionKey`
- `int orgId`

## Returns

- `boolean` - Get the status of crash reporting settings.

## 23.9. METHOD: ISCRASHFILEUPLOADENABLED

Get the status of crash file upload settings for the given organization. Returns true if enabled, false otherwise.

### Parameters

- `string sessionKey`
- `int orgId`

## Returns

- `boolean` - Get the status of crash file upload settings.

## 23.10. METHOD: LISTORGS

Returns the list of organizations.

### Parameters

- `string sessionKey`

### Returns

- `struct organization` info
  - `int id`
  - `string name`
  - `int active_users` - Number of active users in the organization.
  - `int systems` - Number of systems in the organization.
  - `int trusts` - Number of trusted organizations.
  - `int system_groups` - Number of system groups in the organization. (optional)
  - `int activation_keys` - Number of activation keys in the organization. (optional)
  - `int kickstart_profiles` - Number of kickstart profiles in the organization. (optional)
  - `int configuration_channels` - Number of configuration channels in the organization. (optional)
  - `boolean staging_content_enabled` - Is staging content enabled in organization. (optional)

## 23.11. METHOD: LISTSOFTWAREENTITLEMENTS

List software entitlement allocation information across all organizations. Caller must be a satellite administrator.

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct entitlement` usage
    - `string label`
    - `string name`
    - `int free`

- `int used`
- `int allocated`
- `int unallocated`
- `int free_flex`
- `int used_flex`
- `int allocated_flex`
- `int unallocated_flex`

## 23.12. METHOD: LISTSOFTWAREENTITLEMENTS

List each organization's allocation of a given software entitlement. Organizations with no allocation will not be present in the list. A value of -1 indicates unlimited entitlements.

Deprecated - being replaced by `listSoftwareEntitlements(string sessionKey, string label, boolean includeUnentitled)`

### Parameters

- `string sessionKey`
- `string label` - Software entitlement label.

### Returns

- `array`
  - `struct entitlement usage`
    - `int org_id`
    - `string org_name`
    - `int allocated`
    - `int unallocated`
    - `int used`
    - `int free`

## 23.13. METHOD: LISTSOFTWAREENTITLEMENTS

List each organization's allocation of a given software entitlement. A value of -1 indicates unlimited entitlements.

### Parameters

- `string sessionKey`

- string **label** - Software entitlement label.
- boolean **includeUnentitled** - If true, the result will include both organizations that have the entitlement as well as those that do not; otherwise, the result will only include organizations that have the entitlement.

## Returns

- array
  - struct **entitlement** usage
    - int **org\_id**
    - string **org\_name**
    - int **allocated**
    - int **unallocated**
    - int **used**
    - int **free**

Available since: 10.4

## 23.14. METHOD: LISTSOFTWAREENTITLEMENTSFORORG

List an organization's allocation of each software entitlement. A value of -1 indicates unlimited entitlements.

### Parameters

- string **sessionKey**
- int **orgId**

## Returns

- array
  - struct **entitlement** usage
    - string **label**
    - string **name**
    - int **allocated**
    - int **unallocated**
    - int **free**
    - int **used**
    - int **allocated\_flex**

- `int unallocated_flex`
- `int free_flex`
- `int used_flex`

## 23.15. METHOD: LISTSYSTEMENTITLEMENTS

Lists system entitlement allocation information across all organizations. Caller must be a satellite administrator.

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct entitlement usage`
    - `string label`
    - `string name`
    - `int allocated`
    - `int unallocated`
    - `int used`
    - `int free`

## 23.16. METHOD: LISTSYSTEMENTITLEMENTS

List each organization's allocation of a system entitlement. If the organization has no allocation for a particular entitlement, it will not appear in the list.

Deprecated - being replaced by `listSystemEntitlements(string sessionKey, string label, boolean includeUnentitled)`

### Parameters

- `string sessionKey`
- `string label`

### Returns

- `array`
  - `struct entitlement usage`
    - `int org_id`
    - `string org_name`

- `int allocated`
- `int unallocated`
- `int used`
- `int free`

## 23.17. METHOD: LISTSYSTEMENTITLEMENTS

List each organization's allocation of a system entitlement.

### Parameters

- `string sessionKey`
- `string label`
- `boolean includeUnentitled` - If true, the result will include both organizations that have the entitlement as well as those that do not; otherwise, the result will only include organizations that have the entitlement.

### Returns

- `array`
  - `struct entitlement usage`
    - `int org_id`
    - `string org_name`
    - `int allocated`
    - `int unallocated`
    - `int used`
    - `int free`

Available since: 10.4

## 23.18. METHOD: LISTSYSTEMENTITLEMENTSFORORG

List an organization's allocation of each system entitlement.

### Parameters

- `string sessionKey`
- `int orgId`

### Returns

- `array`



- struct entitlement usage

- string label
- string name
- int free
- int used
- int allocated
- int unallocated

## 23.19. METHOD: LISTUSERS

Returns the list of users in a given organization.

### Parameters

- string sessionKey
- int orgId

### Returns

- array
  - struct user
    - string login
    - string login\_uc
    - string name
    - string email
    - boolean is\_org\_admin

## 23.20. METHOD: MIGRATESYSTEMS

Migrate systems from one organization to another. If executed by a Satellite administrator, the systems will be migrated from their current organization to the organization specified by the toOrgId. If executed by an organization administrator, the systems must exist in the same organization as that administrator and the systems will be migrated to the organization specified by the toOrgId. In any scenario, the origination and destination organizations must be defined in a trust.

### Parameters

- string sessionKey
- int toOrgId - ID of the organization where the system(s) will be migrated to.
- array

- `int - systemId`

## Returns

- `array`
  - `int - serverIdMigrated`

## 23.21. METHOD: SETCRASHFILESIZELIMIT

Set the organization wide crash file size limit. The limit value must be non-negative, zero means no limit.

### Parameters

- `string sessionKey`
- `int orgId`
- `int limit` - The limit to set (non-negative value).

### Returns

- `int - 1` on success, exception thrown otherwise.

## 23.22. METHOD: SETCRASHREPORTING

Set the status of crash reporting settings for the given organization. Disabling crash reporting will automatically disable crash file upload.

### Parameters

- `string sessionKey`
- `int orgId`
- `boolean enable` - Use true/false to enable/disable

### Returns

- `int - 1` on success, exception thrown otherwise.

## 23.23. METHOD: SETCRASHFILEUPLOAD

Set the status of crash file upload settings for the given organization. Modifying the settings is possible as long as crash reporting is enabled.

### Parameters

- `string sessionKey`
- `int orgId`
- `boolean enable` - Use true/false to enable/disable

## Returns

- int - 1 on success, exception thrown otherwise.

## 23.24. METHOD: SETPOLICYFORSCAPFILEUPLOAD

Set the status of SCAP detailed result file upload settings for the given organization.

### Parameters

- string `sessionKey`
- int `orgId`
- struct `scap_upload_info`
  - boolean `enabled` - Aggregation of detailed SCAP results is enabled.
  - int `size_limit` - Limit (in Bytes) for a single SCAP file upload.

## Returns

- int - 1 on success, exception thrown otherwise.

## 23.25. METHOD: SETPOLICYFORSCAPRESULTDELETION

Set the status of SCAP result deletion settins for the given organization.

### Parameters

- string `sessionKey`
- int `orgId`
- struct `scap_deletion_info`
  - boolean `enabled` - Deletion of SCAP results is enabled
  - int `retention_period` - Period (in days) after which a scan can be deleted (if enabled).

## Returns

- int - 1 on success, exception thrown otherwise.

## 23.26. METHOD: SETSOFTWAREENTITLEMENTS

Set an organization's entitlement allocation for the given software entitlement. If increasing the entitlement allocation, the default organization (i.e. `orgId=1`) must have a sufficient number of free entitlements.

### Parameters

- string `sessionKey`
- int `orgId`

- `string label` - Software entitlement label.
- `int allocation`

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 23.27. METHOD: SETSOFTWAREFLEXENTITLEMENTS

Set an organization's flex entitlement allocation for the given software entitlement. If increasing the flex entitlement allocation, the default organization (i.e. `orgId=1`) must have a sufficient number of free flex entitlements.

## Parameters

- `string sessionKey`
- `int orgId`
- `string label` - Software entitlement label.
- `int allocation`

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 23.28. METHOD: SETSYSTEMENTITLEMENTS

Set an organization's entitlement allocation for the given software entitlement. If increasing the entitlement allocation, the default organization (i.e. `orgId=1`) must have a sufficient number of free entitlements.

## Parameters

- `string sessionKey`
- `int orgId`
- `string label` - System entitlement label. Valid values include:
  - `enterprise_entitled`
  - `monitoring_entitled`
  - `provisioning_entitled`
  - `virtualization_host`
  - `virtualization_host_platform`
- `int allocation`

## Returns

- `int` - 1 on success, exception thrown otherwise.

- `int` - 1 on success, exception thrown otherwise.

## 23.29. METHOD: UPDATENAME

Updates the name of an organization

### Parameters

- `string sessionKey`
- `int orgId`
- `string name` - Organization name. Must meet same criteria as in the web UI.

### Returns

- `struct organization info`
  - `int id`
  - `string name`
  - `int active_users` - Number of active users in the organization.
  - `int systems` - Number of systems in the organization.
  - `int trusts` - Number of trusted organizations.
  - `int system_groups` - Number of system groups in the organization. (optional)
  - `int activation_keys` - Number of activation keys in the organization. (optional)
  - `int kickstart_profiles` - Number of kickstart profiles in the organization. (optional)
  - `int configuration_channels` - Number of configuration channels in the organization. (optional)
  - `boolean staging_content_enabled` - Is staging content enabled in organization. (optional)

## CHAPTER 24. NAMESPACE: ORG.TRUSTS

Contains methods to access common organization trust information available from the web interface.

### 24.1. METHOD: ADDTRUST

Add an organization to the list of trusted organizations.

#### Parameters

- `string sessionKey`
- `int orgId`
- `int trustOrgId`

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 24.2. METHOD: GETDETAILS

The trust details about an organization given the organization's ID.

#### Parameters

- `string sessionKey`
- `int trustOrgId` - Id of the trusted organization

#### Returns

- `struct org` trust details
  - `dateTime.iso8601 created` - Date the organization was created
  - `dateTime.iso8601 trusted_since` - Date the organization was defined as trusted
  - `int channels_provided` - Number of channels provided by the organization.
  - `int channels_consumed` - Number of channels consumed by the organization.
  - `int systems_migrated_to` - Number of systems migrated to the organization.
  - `int systems_migrated_from` - Number of systems migrated from the organization.

### 24.3. METHOD: LISTCHANNELSCONSUMED

Lists all software channels that the organization given may consume from the user's organization.

#### Parameters

- `string sessionKey`
- `int trustOrgId` - Id of the trusted organization

## Returns

- array
  - struct `channelInfo`
    - int `channel_id`
    - string `channel_name`
    - int `packages`
    - int `systems`

## 24.4. METHOD: LISTCHANNELSPROVIDED

Lists all software channels that the organization given is providing to the user's organization.

### Parameters

- string `sessionKey`
- int `trustOrgId` - Id of the trusted organization

## Returns

- array
  - struct `channelInfo`
    - int `channel_id`
    - string `channel_name`
    - int `packages`
    - int `systems`

## 24.5. METHOD: LISTORGS

List all organizations trusted by the user's organization.

### Parameters

- string `sessionKey`

## Returns

- array
  - struct `trusted organizations`
    - int `org_id`
    - string `org_name`

- `int shared_channels`

## 24.6. METHOD: LISTSYSTEMSAFFECTED

Get a list of systems within the

### Parameters

- `string sessionKey`
- `int orgId`
- `string trustOrgId`

### Returns

- `array`
  - `struct affected systems`
    - `int systemId`
    - `string systemName`

## 24.7. METHOD: LISTTRUSTS

Returns the list of trusted organizations.

### Parameters

- `string sessionKey`
- `int orgId`

### Returns

- `array`
  - `struct trusted organizations`
    - `int orgId`
    - `string orgName`
    - `bool "trustEnabled"`

## 24.8. METHOD: REMOVETRUST

Remove an organization to the list of trusted organizations.

### Parameters

- `string sessionKey`
- `int orgId`



- `int trustOrgId`

## Returns

- `int - 1` on success, exception thrown otherwise.

## CHAPTER 25. NAMESPACE: PACKAGES

Methods to retrieve information about the Packages contained within this server.

### 25.1. METHOD: FINDBYNVREA

Lookup the details for packages with the given name, version, release, architecture label, and (optionally) epoch.

#### Parameters

- `string sessionKey`
- `string name`
- `string version`
- `string release`
- `string epoch` - If set to something other than empty string, strict matching will be used and the epoch string must be correct. If set to an empty string, if the epoch is null or there is only one NVRA combination, it will be returned. (Empty string is recommended.)
- `string archLabel`

#### Returns

- `array`
  - `struct package`
    - `string name`
    - `string version`
    - `string release`
    - `string epoch`
    - `int id`
    - `string arch_label`
    - `string path` - The path on that file system that the package resides
    - `string provider` - The provider of the package, determined by the gpg key it was signed with.
    - `dateTime.iso8601 last_modified`

### 25.2. METHOD: GETDETAILS

Retrieve details for the package with the ID.

#### Parameters

- `string sessionKey`
- `int packageId`

## Returns

- `struct package`
  - `int id`
  - `string name`
  - `string epoch`
  - `string version`
  - `string release`
  - `string arch_label`
  - `array providing_channels`
    - `string Channel` label providing this package.
  - `string build_host`
  - `string description`
  - `string checksum`
  - `string checksum_type`
  - `string vendor`
  - `string summary`
  - `string cookie`
  - `string license`
  - `string file`
  - `string build_date`
  - `string last_modified_date`
  - `string size`
  - `string path` - The path on the Satellite's file system that the package resides.
  - `string payload_size`

## 25.3. METHOD: GETPACKAGE

Retrieve the package file associated with a package. (Consider using `packages.getPackageUrl` for larger files.)

## Parameters

- `string sessionKey`
- `int package_id`

## Returns

- binary object - package file

## 25.4. METHOD: GETPACKAGEURL

Retrieve the url that can be used to download a package. This will expire after a certain time period.

## Parameters

- `string sessionKey`
- `int package_id`

## Returns

- `string` - the download url

## 25.5. METHOD: LISTCHANGELOG

List the change log for a package.

## Parameters

- `string sessionKey`
- `int packageId`

## Returns

- `string`

## 25.6. METHOD: LISTDEPENDENCIES

List the dependencies for a package.

## Parameters

- `string sessionKey`
- `int packageId`

## Returns

array

struct dependency

string dependency

string **dependency\_type** - One of the following:

- requires
- conflicts
- obsoletes
- provides
- recommends
- suggests
- supplements
- enhances

string **dependency\_modifier**

## 25.7. METHOD: LISTFILES

List the files associated with a package.

### Parameters

- string **sessionKey**
- int **packageId**

### Returns

- array
  - struct **file** info
    - string **path**
    - string **type**
    - string **last\_modified\_date**
    - string **checksum**
    - string **checksum\_type**
    - int **size**
    - string **linkto**

## 25.8. METHOD: LISTPROVIDINGCHANNELS

List the channels that provide the a package.

### Parameters

- string **sessionKey**

- `string sessionKey`
- `int packageId`

### Returns

- `array`
  - `struct channel`
    - `string label`
    - `string parent_label`
    - `string name`

## 25.9. METHOD: LISTPROVIDINGERRATA

List the errata providing the a package.

### Parameters

- `string sessionKey`
- `int packageId`

### Returns

- `array`
  - `struct errata`
    - `string advisory`
    - `string issue_date`
    - `string last_modified_date`
    - `string update_date`
    - `string synopsis`
    - `string type`

## 25.10. METHOD: REMOVEPACKAGE

Remove a package from the satellite.

### Parameters

- `string sessionKey`
- `int packageId`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## CHAPTER 26. NAMESPACE: PACKAGES.PROVIDER

Methods to retrieve information about Package Providers associated with packages.

### 26.1. METHOD: ASSOCIATEKEY

Associate a package security key and with the package provider. If the provider or key doesn't exist, it is created. User executing the request must be a Satellite administrator.

#### Parameters

- `string sessionKey`
- `string providerName` - The provider name
- `string key` - The actual key
- `string type` - The type of the key. Currently, only 'gpg' is supported

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 26.2. METHOD: LIST

List all Package Providers. User executing the request must be a Satellite administrator.

#### Parameters

- `string sessionKey`

#### Returns

- `array`
  - `struct package provider`
    - `string name`
    - `array keys`
      - `struct package security key`
        - `string key`
        - `string type`

### 26.3. METHOD: LISTKEYS

List all security keys associated with a package provider. User executing the request must be a Satellite administrator.

#### Parameters

- `string sessionKey`



- string `providerName` - The provider name

## Returns

- array
  - struct `package` security key
    - string `key`
    - string `type`

## CHAPTER 27. NAMESPACE: PACKAGES.SEARCH

Methods to interface to package search capabilities in search server..

### 27.1. METHOD: ADVANCED

Advanced method to search lucene indexes with a passed in query written in Lucene Query Parser syntax. Lucene Query Parser syntax is defined at

#### Parameters

- string `sessionKey`
- string `luceneQuery` - a query written in the form of Lucene QueryParser Syntax

#### Returns

- array
  - struct `package overview`
    - int `id`
    - string `name`
    - string `summary`
    - string `description`
    - string `version`
    - string `release`
    - string `arch`
    - string `epoch`
    - string `provider`

### 27.2. METHOD: ADVANCEDWITHACTKEY

Advanced method to search lucene indexes with a passed in query written in Lucene Query Parser syntax, additionally this method will limit results to those which are associated with a given activation key. Lucene Query Parser syntax is defined at

#### Parameters

- string `sessionKey`
- string `luceneQuery` - a query written in the form of Lucene QueryParser Syntax
- string `actKey` - activation key to look for packages in

#### Returns

- array
  - struct **package** overview
    - int **id**
    - string **name**
    - string **summary**
    - string **description**
    - string **version**
    - string **release**
    - string **arch**
    - string **epoch**
    - string **provider**

### 27.3. METHOD: ADVANCEDWITHCHANNEL

Advanced method to search lucene indexes with a passed in query written in Lucene Query Parser syntax, additionally this method will limit results to those which are in the passed in channel label. Lucene Query Parser syntax is defined at

#### Parameters

- string **sessionKey**
- string **luceneQuery** - a query written in the form of Lucene QueryParser Syntax
- string **channelLabel** - Channel Label

#### Returns

- array
  - struct **package** overview
    - int **id**
    - string **name**
    - string **summary**
    - string **description**
    - string **version**
    - string **release**
    - string **arch**

- `string epoch`
- `string provider`

## 27.4. METHOD: NAME

Search the lucene package indexes for all packages which match the given name.

### Parameters

- `string sessionKey`
- `string name` - package name to search for

### Returns

- `array`
  - `struct package overview`
    - `int id`
    - `string name`
    - `string summary`
    - `string description`
    - `string version`
    - `string release`
    - `string arch`
    - `string epoch`
    - `string provider`

## 27.5. METHOD: NAMEANDDESCRIPTION

Search the lucene package indexes for all packages which match the given query in name or description

### Parameters

- `string sessionKey`
- `string query` - text to match in package name or description

### Returns

- `array`
  - `struct package overview`
    - `int id`

- `string name`
- `string summary`
- `string description`
- `string version`
- `string release`
- `string arch`
- `string epoch`
- `string provider`

## 27.6. METHOD: NAMEANDSUMMARY

Search the lucene package indexes for all packages which match the given query in name or summary.

### Parameters

- `string sessionKey`
- `string query` - text to match in package name or summary

### Returns

- `array`
  - `struct package overview`
    - `int id`
    - `string name`
    - `string summary`
    - `string description`
    - `string version`
    - `string release`
    - `string arch`
    - `string epoch`
    - `string provider`

## CHAPTER 28. NAMESPACE: PREFERENCES.LOCALE

Provides methods to access and modify user locale information

### 28.1. METHOD: LISTLOCALES

Returns a list of all understood locales. Can be used as input to `setLocale`.

#### Parameters

#### Returns

- array
  - string - Locale code.

### 28.2. METHOD: LISTTIMEZONES

Returns a list of all understood timezones. Results can be used as input to `setTimeZone`.

#### Parameters

#### Returns

- array
  - struct `timezone`
    - int `time_zone_id` - Unique identifier for timezone.
    - string `olson_name` - Name as identified by the Olson database.

### 28.3. METHOD: SETLOCALE

Set a user's locale.

#### Parameters

- string `sessionKey`
- string `login` - User's login name.
- string `locale` - Locale to set. (from `listLocales`)

#### Returns

- int - 1 on success, exception thrown otherwise.

### 28.4. METHOD: SETTIMEZONE

Set a user's timezone.

#### Parameters

- string `sessionKey`

- `string login` - User's login name.
- `int tzid` - Timezone ID. (from `listTimeZones`)

## Returns

- `int` - 1 on success, exception thrown otherwise.

## CHAPTER 29. NAMESPACE: PROXY

Provides methods to activate/deactivate a proxy server.

### 29.1. METHOD: ACTIVATEPROXY

Activates the proxy identified by the given client certificate i.e. systemid file.

#### Parameters

- string `systemid` - systemid file
- string `version` - Version of proxy to be registered.

#### Returns

- int - 1 on success, exception thrown otherwise.

### 29.2. METHOD: CREATEMONITORINGSCOUT

Create Monitoring Scout for proxy.

#### Parameters

- string `systemid` - systemid file

#### Returns

- string

Available since: 10.7

### 29.3. METHOD: DEACTIVATEPROXY

Deactivates the proxy identified by the given client certificate i.e. systemid file.

#### Parameters

- string `systemid` - systemid file

#### Returns

- int - 1 on success, exception thrown otherwise.

### 29.4. METHOD: ISPROXY

Test, if the system identified by the given client certificate i.e. systemid file, is proxy.

#### Parameters

- string `systemid` - systemid file

#### Returns

- int - 1 on success, exception thrown otherwise.



## 29.5. METHOD: LISTAVAILABLEPROXYCHANNELS

List available version of proxy channel for system identified by the given client certificate i.e. systemid file.

### Parameters

- string `systemid` - systemid file

### Returns

- array
  - string - version

Available since: 10.5

## CHAPTER 30. NAMESPACE: SATELLITE

Provides methods to obtain details on the Satellite.

### 30.1. METHOD: GETCERTIFICATEEXPIRATIONDATE

Retrieves the certificate expiration date of the activated certificate.

#### Parameters

- `string sessionKey`

#### Returns

- `dateTime.iso8601`

### 30.2. METHOD: ISMONITORINGENABLED

Indicates if monitoring is enabled on the satellite

#### Parameters

- `string sessionKey`

#### Returns

- `boolean True` if monitoring is enabled

### 30.3. METHOD: ISMONITORINGENABLEDBYSYSTEMID

Indicates if monitoring is enabled on the satellite

#### Parameters

- `string systemid` - systemid file

#### Returns

- `boolean True` if monitoring is enabled

### 30.4. METHOD: LISTENTITLEMENTS

Lists all channel and system entitlements for the organization associated with the user executing the request.

#### Parameters

- `string sessionKey`

#### Returns

- `struct channel/system entitlements`
  - `array system`

- struct `system` entitlement
  - string `label`
  - string `name`
  - int `used_slots`
  - int `free_slots`
  - int `total_slots`
- array `channel`
  - struct `channel` entitlement
    - string `label`
    - string `name`
    - int `used_slots`
    - int `free_slots`
    - int `total_slots`
    - int `used_flex`
    - int `free_flex`
    - int `total_flex`

## 30.5. METHOD: LISTPROXIES

List the proxies within the user's organization.

### Parameters

- string `sessionKey`

### Returns

- array
  - struct `system`
    - int `id`
    - string `name`
    - dateTime.iso8601 `last_checkin` - Last time server successfully checked in

## CHAPTER 31. NAMESPACE: SCHEDULE

Methods to retrieve information about scheduled actions.

### 31.1. METHOD: ARCHIVEACTIONS

Archive all actions in the given list.

#### Parameters

- `string sessionKey`
- `array`
  - `int` - action id

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 31.2. METHOD: CANCELACTIONS

Cancel all actions in given list. If an invalid action is provided, none of the actions given will canceled.

#### Parameters

- `string sessionKey`
- `array`
  - `int` - action id

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 31.3. METHOD: DELETEACTIONS

Delete all archived actions in the given list.

#### Parameters

- `string sessionKey`
- `array`
  - `int` - action id

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 31.4. METHOD: LISTALLACTIONS

Returns a list of all actions. This includes completed, in progress, failed and archived actions.

## Parameters

- `string sessionKey`

## Returns

- `array`
  - `struct action`
    - `int id` - Action Id.
    - `string name` - Action name.
    - `string type` - Action type.
    - `string scheduler` - The user that scheduled the action. (optional)
    - `dateTime.iso8601 earliest` - The earliest date and time the action will be performed
    - `int completedSystems` - Number of systems that completed the action.
    - `int failedSystems` - Number of systems that failed the action.
    - `int inProgressSystems` - Number of systems that are in progress.

## 31.5. METHOD: LISTARCHIVEDACTIONS

Returns a list of actions that have been archived.

## Parameters

- `string sessionKey`

## Returns

- `array`
  - `struct action`
    - `int id` - Action Id.
    - `string name` - Action name.
    - `string type` - Action type.
    - `string scheduler` - The user that scheduled the action. (optional)
    - `dateTime.iso8601 earliest` - The earliest date and time the action will be performed
    - `int completedSystems` - Number of systems that completed the action.
    - `int failedSystems` - Number of systems that failed the action.
    - `int inProgressSystems` - Number of systems that are in progress.

## 31.6. METHOD: LISTCOMPLETEDACTIONS

Returns a list of actions that have completed successfully.

### Parameters

- string `sessionKey`

### Returns

- array
  - struct `action`
    - int `id` - Action Id.
    - string `name` - Action name.
    - string `type` - Action type.
    - string `scheduler` - The user that scheduled the action. (optional)
    - dateTime.iso8601 `earliest` - The earliest date and time the action will be performed
    - int `completedSystems` - Number of systems that completed the action.
    - int `failedSystems` - Number of systems that failed the action.
    - int `inProgressSystems` - Number of systems that are in progress.

## 31.7. METHOD: LISTCOMPLETEDSYSTEMS

Returns a list of systems that have completed a specific action.

### Parameters

- string `sessionKey`
- string `actionId`

### Returns

- array
  - struct `system`
    - int `server_id`
    - string `server_name` - Server name.
    - string `base_channel` - Base channel used by the server.
    - dateTime.iso8601 `timestamp` - The time the action was completed
    - string `message` - Optional message containing details on the execution of the action. For example, if the action failed, this will contain the failure text.

## 31.8. METHOD: LISTFAILED ACTIONS

Returns a list of actions that have failed.

### Parameters

- string `sessionKey`

### Returns

- array
  - struct `action`
    - int `id` - Action Id.
    - string `name` - Action name.
    - string `type` - Action type.
    - string `scheduler` - The user that scheduled the action. (optional)
    - `dateTime.iso8601` `earliest` - The earliest date and time the action will be performed
    - int `completedSystems` - Number of systems that completed the action.
    - int `failedSystems` - Number of systems that failed the action.
    - int `inProgressSystems` - Number of systems that are in progress.

## 31.9. METHOD: LISTFAILED SYSTEMS

Returns a list of systems that have failed a specific action.

### Parameters

- string `sessionKey`
- string `actionId`

### Returns

- array
  - struct `system`
    - int `server_id`
    - string `server_name` - Server name.
    - string `base_channel` - Base channel used by the server.
    - `dateTime.iso8601` `timestamp` - The time the action was completed
    - string `message` - Optional message containing details on the execution of the action. For example, if the action failed, this will contain the failure text.

## 31.10. METHOD: LISTINPROGRESSACTIONS

Returns a list of actions that are in progress.

### Parameters

- string `sessionKey`

### Returns

- array
  - struct `action`
    - int `id` - Action Id.
    - string `name` - Action name.
    - string `type` - Action type.
    - string `scheduler` - The user that scheduled the action. (optional)
    - dateTime.iso8601 `earliest` - The earliest date and time the action will be performed
    - int `completedSystems` - Number of systems that completed the action.
    - int `failedSystems` - Number of systems that failed the action.
    - int `inProgressSystems` - Number of systems that are in progress.

## 31.11. METHOD: LISTINPROGRESSSYSTEMS

Returns a list of systems that have a specific action in progress.

### Parameters

- string `sessionKey`
- string `actionId`

### Returns

- array
  - struct `system`
    - int `server_id`
    - string `server_name` - Server name.
    - string `base_channel` - Base channel used by the server.
    - dateTime.iso8601 `timestamp` - The time the action was completed
    - string `message` - Optional message containing details on the execution of the action. For example, if the action failed, this will contain the failure text.



## 31.12. METHOD: RESCHEDULEACTIONS

Reschedule all actions in the given list.

### Parameters

- string `sessionKey`
- array
  - int - action id
- boolean `onlyFailed` - True to only reschedule failed actions, False to reschedule all

### Returns

- int - 1 on success, exception thrown otherwise.

## CHAPTER 32. NAMESPACE: SYNC.MASTER

Contains methods to set up information about known-"masters", for use on the "slave" side of ISS

### 32.1. METHOD: ADDTOMASTER

Add a single organizations to the list of those the specified Master has exported to this Slave

#### Parameters

- `string sessionKey`
- `int id` - Id of the desired Master
- `struct master-org` details
  - `int masterOrgId`
  - `string masterOrgName`
  - `int localOrgId`

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 32.2. METHOD: CREATE

Create a new Master, known to this Slave.

#### Parameters

- `string sessionKey`
- `string label` - Master's fully-qualified domain name

#### Returns

- `struct IssMaster` info
  - `int id`
  - `string label`
  - `string caCert`
  - `boolean isCurrentMaster`

### 32.3. METHOD: DELETE

Remove the specified Master

#### Parameters

- `string sessionKey`

- `int id` - Id of the Master to remove

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 32.4. METHOD: GETDEFAULTMASTER

Return the current default-Master for this Slave

### Parameters

- `string sessionKey`

### Returns

- `struct IssMaster` info
  - `int id`
  - `string label`
  - `string caCert`
  - `boolean isCurrentMaster`

## 32.5. METHOD: GETMASTER

Find a Master by specifying its ID

### Parameters

- `string sessionKey`
- `int id` - Id of the desired Master

### Returns

- `struct IssMaster` info
  - `int id`
  - `string label`
  - `string caCert`
  - `boolean isCurrentMaster`

## 32.6. METHOD: GETMASTERBYLABEL

Find a Master by specifying its label

### Parameters

- `string sessionKey`

- string `label` - Label of the desired Master

## Returns

- struct `IssMaster` info
  - int `id`
  - string `label`
  - string `caCert`
  - boolean `isCurrentMaster`

## 32.7. METHOD: GETMASTERORGS

List all organizations the specified Master has exported to this Slave

### Parameters

- string `sessionKey`
- int `id` - Id of the desired Master

## Returns

- array
  - struct `IssMasterOrg` info
    - int `masterOrgId`
    - string `masterOrgName`
    - int `localOrgId`

## 32.8. METHOD: GETMASTERS

Get all the Masters this Slave knows about

### Parameters

- string `sessionKey`

## Returns

- array
  - struct `IssMaster` info
    - int `id`
    - string `label`
    - string `caCert`

- boolean `isCurrentMaster`

## 32.9. METHOD: MAKEDEFAULT

Make the specified Master the default for this Slave's satellite-sync

### Parameters

- string `sessionKey`
- int `id` - Id of the Master to make the default

### Returns

- int - 1 on success, exception thrown otherwise.

## 32.10. METHOD: MAPTOLOCAL

Add a single organizations to the list of those the specified Master has exported to this Slave

### Parameters

- string `sessionKey`
- int `masterId` - Id of the desired Master
- int `masterOrgId` - Id of the desired Master
- int `localOrgId` - Id of the desired Master

### Returns

- int - 1 on success, exception thrown otherwise.

## 32.11. METHOD: SETCACERT

Set the CA-CERT filename for specified Master on this Slave

### Parameters

- string `sessionKey`
- int `id` - Id of the Master to affect
- string `caCertFilename` - path to specified Master's CA cert

### Returns

- int - 1 on success, exception thrown otherwise.

## 32.12. METHOD: SETMASTERORGS

Reset all organizations the specified Master has exported to this Slave

## Parameters

- `string sessionKey`
- `int id` - Id of the desired Master
- `array`
  - `struct master - org details`
    - `int masterOrgId`
    - `string masterOrgName`
    - `int localOrgId`

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 32.13. METHOD: UNSETDEFAULTMASTER

Make this slave have no default Master for satellite-sync

### Parameters

- `string sessionKey`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 32.14. METHOD: UPDATE

Updates the label of the specified Master

### Parameters

- `string sessionKey`
- `int id` - Id of the Master to update
- `string label` - Desired new label

### Returns

- `struct IssMaster` info
  - `int id`
  - `string label`
  - `string caCert`
  - `boolean isCurrentMaster`

## CHAPTER 33. NAMESPACE: SYNC.SLAVE

Contains methods to set up information about allowed-"slaves", for use on the "master" side of ISS

### 33.1. METHOD: CREATE

Create a new Slave, known to this Master.

#### Parameters

- string `sessionKey`
- string `slave` - Slave's fully-qualified domain name
- boolean `enabled` - Let this slave talk to us?
- boolean `allowAllOrgs` - Export all our orgs to this slave?

#### Returns

- struct `IssSlave` info
  - int `id`
  - string `slave`
  - boolean `enabled`
  - boolean `allowAllOrgs`

### 33.2. METHOD: DELETE

Remove the specified Slave

#### Parameters

- string `sessionKey`
- int `id` - Id of the Slave to remove

#### Returns

- int - 1 on success, exception thrown otherwise.

### 33.3. METHOD: GETALLOWEDORGS

Get all orgs this Master is willing to export to the specified Slave

#### Parameters

- string `sessionKey`
- int `id` - Id of the desired Slave

## Returns

- array
  - int - ids of allowed organizations

## 33.4. METHOD: GETSLAVE

Find a Slave by specifying its ID

### Parameters

- string `sessionKey`
- int `id` - Id of the desired Slave

### Returns

- struct `IssSlave` info
  - int `id`
  - string `slave`
  - boolean `enabled`
  - boolean `allowAllOrgs`

## 33.5. METHOD: GETSLAVEBYNAME

Find a Slave by specifying its Fully-Qualified Domain Name

### Parameters

- string `sessionKey`
- string `fqdn` - Domain-name of the desired Slave

### Returns

- struct `IssSlave` info
  - int `id`
  - string `slave`
  - boolean `enabled`
  - boolean `allowAllOrgs`

## 33.6. METHOD: GETSLAVES

Get all the Slaves this Master knows about

### Parameters



- `string sessionKey`

## Returns

- `array`
  - `struct IssSlave info`
    - `int id`
    - `string slave`
    - `boolean enabled`
    - `boolean allowAllOrgs`

## 33.7. METHOD: SETALLOWEDORGS

Set the orgs this Master is willing to export to the specified Slave

### Parameters

- `string sessionKey`
- `int id` - Id of the desired Slave
- `array`
  - `int` - List of org-ids we're willing to export

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 33.8. METHOD: UPDATE

Updates attributes of the specified Slave

### Parameters

- `string sessionKey`
- `int id` - Id of the Slave to update
- `string slave` - Slave's fully-qualified domain name
- `boolean enabled` - Let this slave talk to us?
- `boolean allowAllOrgs` - Export all our orgs to this Slave?

### Returns

- `struct IssSlave info`
  - `int id`

- **string slave**
- **boolean enabled**
- **boolean allowAllOrgs**

## CHAPTER 34. NAMESPACE: SYSTEM

Provides methods to access and modify registered system.

### 34.1. METHOD: ADDENTITLEMENTS

Add entitlements to a server. Entitlements a server already has are quietly ignored.

#### Parameters

- `string sessionKey`
- `int serverId`
- `array`
  - `string` - `entitlementLabel` - one of following: `monitoring_entitled`, `provisioning_entitled`, `virtualization_host`, `virtualization_host_platform`, `enterprise_entitled`

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 34.2. METHOD: ADDNOTE

Add a new note to the given server.

#### Parameters

- `string sessionKey`
- `int serverId`
- `string subject` - What the note is about.
- `string body` - Content of the note.

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 34.3. METHOD: APPLYERRATA

Schedules an action to apply errata updates to a system.

Deprecated - being replaced by `system.scheduleApplyErrata(string sessionKey, int serverId, array[int errataId])`

#### Parameters

- `string sessionKey`
- `int serverId`
- `array`

- `int - errataId`

## Returns

- `int - 1` on success, exception thrown otherwise.

## 34.4. METHOD: COMPAREPACKAGEPROFILE

Compare a system's packages against a package profile. In the result returned, 'this\_system' represents the server provided as an input and 'other\_system' represents the profile provided as an input.

### Parameters

- `string sessionKey`
- `int serverId`
- `string profileLabel`

### Returns

array

struct **Package Metadata**

`int package_name_id`

`string package_name`

`string package_arch`

`string this_system` - Version of package on this system.

`string other_system` - Version of package on the other system.

`int comparison`

- `0` - No difference.
- `1` - Package on this system only.
- `2` - Newer package version on this system.
- `3` - Package on other system only.
- `4` - Newer package version on other system.

## 34.5. METHOD: COMPAREPACKAGES

Compares the packages installed on two systems.

### Parameters

- `string sessionKey`
- `int thisServerId`

- `int otherServerId`

## Returns

array

struct **Package Metadata**

`int package_name_id`

`string package_name`

`string package_arch`

`string this_system` - Version of package on this system.

`string other_system` - Version of package on the other system.

`int comparison`

- 0 - No difference.
- 1 - Package on this system only.
- 2 - Newer package version on this system.
- 3 - Package on other system only.
- 4 - Newer package version on other system.

## 34.6. METHOD: CONVERTTOFLEXENTITLEMENT

Converts the given list of systems for a given channel family to use the flex entitlement.

### Parameters

- `string sessionKey`
- array
  - `int` - `serverId`
- `string channelFamilyLabel`

### Returns

- `int` - the total the number of systems that were converted to use flex entitlement.

## 34.7. METHOD: CREATEPACKAGEPROFILE

Create a new stored Package Profile from a systems installed package list.

### Parameters

- `string sessionKey`
- `int serverId`

- `string profileLabel`
- `string description`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.8. METHOD: CREATESYSTEMRECORD

Creates a cobbler system record with the specified kickstart label

### Parameters

- `string sessionKey`
- `int serverId`
- `string ksLabel`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.9. METHOD: DELETECUSTOMVALUES

Delete the custom values defined for the custom system information keys provided from the given system.

### Parameters

- `string sessionKey`
- `int serverId`
- `array`
  - `string` - `customInfoLabel`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.10. METHOD: DELETEGUESTPROFILES

Delete the specified list of guest profiles for a given host

### Parameters

- `string sessionKey`
- `int hostId`
- `array`

- `string` - `guestNames`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.11. METHOD: DELETENOTE

Deletes the given note from the server.

### Parameters

- `string` `sessionKey`
- `int` `serverId`
- `int` `noteId`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.12. METHOD: DELETENOTES

Deletes all notes from the server.

### Parameters

- `string` `sessionKey`
- `int` `serverId`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.13. METHOD: DELETEPACKAGEPROFILE

Delete a package profile

### Parameters

- `string` `sessionKey`
- `int` `profileId`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.14. METHOD: DELETESYSTEM

Delete a system given its client certificate.

### Parameters

- string `systemid` - systemid file

## Returns

- int - 1 on success, exception thrown otherwise.

Available since: 10.10

## 34.15. METHOD: DELETESYSTEM

Delete a system given its server id synchronously

### Parameters

- string `sessionKey`
- int `serverId`

## Returns

- int - 1 on success, exception thrown otherwise.

## 34.16. METHOD: DELETESYSTEMS

Delete systems given a list of system ids asynchronously.

### Parameters

- string `sessionKey`
- array
  - int - `serverId`

## Returns

- int - 1 on success, exception thrown otherwise.

## 34.17. METHOD: DELETETAGFROMSNAPSHOT

Deletes tag from system snapshot

### Parameters

- string `sessionKey`
- int `serverId`
- string `tagName`

## Returns

- int - 1 on success, exception thrown otherwise.

## 34.18. METHOD: DOWNLOADSYSTEMID



Get the system ID file for a given server.

### Parameters

- string `sessionKey`
- int `serverId`

### Returns

- string

## 34.19. METHOD: GETCONNECTIONPATH

Get the list of proxies that the given system connects through in order to reach the server.

### Parameters

- string `sessionKey`
- int `serverId`

### Returns

- array
  - struct proxy connection path details
    - int `position` - Position of proxy in chain. The proxy that the system connects directly to is listed in position 1.
    - int `id` - Proxy system id
    - string `hostname` - Proxy host name

## 34.20. METHOD: GETCPU

Gets the CPU information of a system.

### Parameters

- string `sessionKey`
- int `serverId`

### Returns

- struct CPU
  - string `cache`
  - string `family`
  - string `mhz`
  - string `flags`

- `string model`
- `string vendor`
- `string arch`
- `string stepping`
- `string count`
- `int socket_count` (if available)"

## 34.21. METHOD: GETCUSTOMVALUES

Get the custom data values defined for the server.

### Parameters

- `string sessionKey`
- `int serverId`

### Returns

- `struct custom value`
  - `string custom info label"`

## 34.22. METHOD: GETDETAILS

Get system details.

### Parameters

- `string sessionKey`
- `int serverId`

### Returns

- `struct server details`
  - `int id` - System id
  - `string profile_name`
  - `string base_entitlement` - System's base entitlement label. (enterprise\_entitled or sw\_mgr\_entitled)
  - `array string`
    - `addon_entitlements` System's addon entitlements labels, including `monitoring_entitled`, `provisioning_entitled`, `virtualization_host`, `virtualization_host_platform`

- boolean **auto\_update** - True if system has auto errata updates enabled.
- string **release** - The Operating System release (i.e. 4AS, 5Server)
- string **address1**
- string **address2**
- string **city**
- string **state**
- string **country**
- string **building**
- string **room**
- string **rack**
- string **description**
- string **hostname**
- dateTime.iso8601 **last\_boot**
- string **osa\_status** - Either 'unknown', 'offline', or 'online'.
- boolean **lock\_status** - True indicates that the system is locked. False indicates that the system is unlocked.
- string **virtualization** - Virtualization type - for virtual guests only (optional)

### 34.23. METHOD: GETDEVICES

Gets a list of devices for a system.

#### Parameters

- string **sessionKey**
- int **serverId**

#### Returns

- array
  - struct **device**
    - string **device** - optional
    - string **device\_class** - Includes CDROM, FIREWIRE, HD, USB, VIDEO, OTHER, etc.
    - string **driver**
    - string **description**

- `string bus`
- `string pcitype`

## 34.24. METHOD: GETDMI

Gets the DMI information of a system.

### Parameters

- `string sessionKey`
- `int serverId`

### Returns

- `struct DMI`
  - `string vendor`
  - `string system`
  - `string product`
  - `string asset`
  - `string board`
  - `string bios_release` - (optional)
  - `string bios_vendor` - (optional)
  - `string bios_version` - (optional)

## 34.25. METHOD: GETENTITLEMENTS

Gets the entitlements for a given server.

### Parameters

- `string sessionKey`
- `int serverId`

### Returns

- `array`
  - `string` - `entitlement_label`

## 34.26. METHOD: GETEVENTHISTORY

Returns a list history items associated with the system, ordered from newest to oldest. Note that the details may be empty for events that were scheduled against the system (as compared to instant). For more information on such events, see the `system.listSystemEvents` operation.

## Parameters

- `string sessionKey`
- `int serverId`

## Returns

- `array`
  - `struct HistoryEvent`
    - `dateTime.iso8601 completed` - Date that the event occurred (optional)
    - `string summary` - Summary of the event
    - `string details` - Details of the event

## 34.27. METHOD: GETID

Get system IDs and last check in information for the given system name.

### Parameters

- `string sessionKey`
- `string systemName`

### Returns

- `array`
  - `struct system`
    - `int id`
    - `string name`
    - `dateTime.iso8601 last_checkin` - Last time server successfully checked in

## 34.28. METHOD: GETMEMORY

Gets the memory information for a system.

### Parameters

- `string sessionKey`
- `int serverId`

### Returns

- `struct memory`
  - `int ram` - The amount of physical memory in MB.

- `int swap` - The amount of swap space in MB.

## 34.29. METHOD: GETNAME

Get system name and last check in information for the given system ID.

### Parameters

- `string sessionKey`
- `string serverId`

### Returns

- `struct name` info
  - `int id` - Server id
  - `string name` - Server name
  - `dateTime.iso8601 last_checkin` - Last time server successfully checked in

## 34.30. METHOD: GETNETWORK

Get the addresses and hostname for a given server.

### Parameters

- `string sessionKey`
- `int serverId`

### Returns

- `struct network` info
  - `string ip` - IPv4 address of server
  - `string ip6` - IPv6 address of server
  - `string hostname` - Hostname of server

## 34.31. METHOD: GETNETWORKDEVICES

Returns the network devices for the given server.

### Parameters

- `string sessionKey`
- `int serverId`

### Returns

array

**struct network device**

**string ip** - IP address assigned to this network device

**string interface** - Network interface assigned to device e.g. eth0

**string netmask** - Network mask assigned to device

**string hardware\_address** - Hardware Address of device.

**string module** - Network driver used for this device.

**string broadcast** - Broadcast address for device.

**array ipv6** - List of IPv6 addresses

**array**

- **struct ipv6 address**
  - **string address** - IPv6 address of this network device
  - **string netmask** - IPv6 netmask of this network device
  - **string scope** - IPv6 address scope

## 34.32. METHOD: GETREGISTRATIONDATE

Returns the date the system was registered.

### Parameters

- **string sessionKey**
- **int serverId**

### Returns

- **dateTime.iso8601** - The date the system was registered, in local time.

## 34.33. METHOD: GETRELEVANTERRATA

Returns a list of all errata that are relevant to the system.

### Parameters

- **string sessionKey**
- **int serverId**

### Returns

- **array**
  - **struct errata**

- `int id` - Errata ID.
- `string date` - Date erratum was created.
- `string update_date` - Date erratum was updated.
- `string advisory_synopsis` - Summary of the erratum.
- `string advisory_type` - Type label such as Security, Bug Fix
- `string advisory_name` - Name such as RHSA, etc

### 34.34. METHOD: GETRELEVANTERRATABYTYPE

Returns a list of all errata of the specified type that are relevant to the system.

#### Parameters

- `string sessionKey`
- `int serverId`
- `string advisoryType` - type of advisory (one of of the following: 'Security Advisory', 'Product Enhancement Advisory', 'Bug Fix Advisory')

#### Returns

- `array`
  - `struct errata`
    - `int id` - Errata ID.
    - `string date` - Date erratum was created.
    - `string update_date` - Date erratum was updated.
    - `string advisory_synopsis` - Summary of the erratum.
    - `string advisory_type` - Type label such as Security, Bug Fix
    - `string advisory_name` - Name such as RHSA, etc

### 34.35. METHOD: GETRUNNINGKERNEL

Returns the running kernel of the given system.

#### Parameters

- `string sessionKey`
- `int serverId`

#### Returns



- `string`

### 34.36. METHOD: GETSCRIPTACTIONDETAILS

Returns script details for script run actions

#### Parameters

- `string sessionKey`
- `int actionId` - ID of the script run action.

#### Returns

`struct Script` details

`int id` - action id

`string content` - script content

`string run_as_user` - Run as user

`string run_as_group` - Run as group

`int timeout` - Timeout in seconds

`array`

- `struct script` result
  - `int serverId` - ID of the server the script runs on.
  - `dateTime.iso8601 startDate` - Time script began execution.
  - `dateTime.iso8601 stopDate` - Time script stopped execution.
  - `int returnCode` - Script execution return code.
  - `string output` - Output of the script (base64 encoded according to the `output_enc64` attribute)
  - `boolean output_enc64` - Identifies base64 encoded output

- result

### 34.37. METHOD: GETSCRIPTRESULTS

Fetch results from a script execution. Returns an empty array if no results are yet available.

#### Parameters

- `string sessionKey`
- `int actionId` - ID of the script run action.

#### Returns

- **array**
  - **struct script result**
    - **int serverId** - ID of the server the script runs on.
    - **dateTime.iso8601 startDate** - Time script began execution.
    - **dateTime.iso8601 stopDate** - Time script stopped execution.
    - **int returnCode** - Script execution return code.
    - **string output** - Output of the script (base64 encoded according to the output\_enc64 attribute)
    - **boolean output\_enc64** - Identifies base64 encoded output

## 34.38. METHOD: GETSUBSCRIBEDBASECHANNEL

Provides the base channel of a given system

### Parameters

- **string sessionKey**
- **int serverId**

### Returns

**struct channel**

**int id**

**string name**

**string label**

**string arch\_name**

**string arch\_label**

**string summary**

**string description**

**string checksum\_label**

**dateTime.iso8601 last\_modified**

**string maintainer\_name**

**string maintainer\_email**

**string maintainer\_phone**

**string support\_policy**

string `gpg_key_url`

string `gpg_key_id`

string `gpg_key_fp`

dateTime.iso8601 `yumrepo_last_sync` - (optional)

string `end_of_life`

string `parent_channel_label`

string `clone_original`

array

- struct `contentSources`
  - int `id`
  - string `label`
  - string `sourceUrl`
  - string `type`

### 34.39. METHOD: GETSYSTEMCURRENCYMULTIPLIERS

Get the System Currency score multipliers

#### Parameters

- string `sessionKey`

#### Returns

- Map of score multipliers

### 34.40. METHOD: GETSYSTEMCURRENCYSCORES

Get the System Currency scores for all servers the user has access to

#### Parameters

- string `sessionKey`

#### Returns

- array
  - struct `system currency`
    - int `sid`
    - int `critical security errata count`

- `int important security errata count"`
- `int moderate security errata count"`
- `int low security errata count"`
- `int bug fix errata count"`
- `int enhancement errata count"`
- `int system currency score"`

### 34.41. METHOD: GETUNSCHEDULEDERRATA

Provides an array of errata that are applicable to a given system.

#### Parameters

- `string sessionKey`
- `int serverId`

#### Returns

- `array`
  - `struct errata`
    - `int id` - Errata Id
    - `string date` - Date erratum was created.
    - `string advisory_type` - Type of the advisory.
    - `string advisory_name` - Name of the advisory.
    - `string advisory_synopsis` - Summary of the erratum.

### 34.42. METHOD: GETUUID

Get the UUID from the given system ID.

#### Parameters

- `string sessionKey`
- `int serverId`

#### Returns

- `string`

### 34.43. METHOD: GETVARIABLES

Lists kickstart variables set in the system record for the specified server. Note: This call assumes that

a system record exists in cobbler for the given system and will raise an XMLRPC fault if that is not the case. To create a system record over xmlrpc use `system.createSystemRecord` To create a system record in the Web UI please go to System - Specified System - Provisioning - Select a Kickstart profile - Create Cobbler System Record.

### Parameters

- `string sessionKey`
- `int serverId`

### Returns

- `struct System` kickstart variables
  - `boolean netboot` - netboot enabled
  - `array kickstart variables"`
    - `struct kickstart variable`
      - `string key`
      - `string or int "value"`

## 34.44. METHOD: ISNVREINSTALLED

Check if the package with the given NVRE is installed on given system.

### Parameters

- `string sessionKey`
- `int serverId`
- `string name` - Package name.
- `string version` - Package version.
- `string release` - Package release.

### Returns

- 1 if package exists, 0 if not, exception is thrown if an error occurs

## 34.45. METHOD: ISNVREINSTALLED

Is the package with the given NVRE installed on given system.

### Parameters

- `string sessionKey`
- `int serverId`
- `string name` - Package name.

- string **version** - Package version.
- string **release** - Package release.
- string **epoch** - Package epoch.

## Returns

- 1 if package exists, 0 if not, exception is thrown if an error occurs

## 34.46. METHOD: LISTACTIVATIONKEYS

List the activation keys the system was registered with. An empty list will be returned if an activation key was not used during registration.

## Parameters

- string **sessionKey**
- int **serverId**

## Returns

- array
  - string - key

## 34.47. METHOD: LISTACTIVESYSTEMS

Returns a list of active servers visible to the user.

## Parameters

- string **sessionKey**

## Returns

- array
  - struct **system**
    - int **id**
    - string **name**
    - dateTime.iso8601 **last\_checkin** - Last time server successfully checked in

## 34.48. METHOD: LISTACTIVESYSTEMSDETAILS

Given a list of server ids, returns a list of active servers' details visible to the user.

## Parameters

- string **sessionKey**

- **array**
  - **int** - `serverId`s

## Returns

**array**

**struct server** details

**int** `id` - The server's id

**string** `name` - The server's name

**dateTime.iso8601** `last_checkin` - Last time server successfully checked in (in UTC)

**int** `ram` - The amount of physical memory in MB.

**int** `swap` - The amount of swap space in MB.

**struct network\_devices** The server's network devices

**struct network** device

**string** `ip` - IP address assigned to this network device

**string** `interface` - Network interface assigned to device e.g. `eth0`

**string** `netmask` - Network mask assigned to device

**string** `hardware_address` - Hardware Address of device.

**string** `module` - Network driver used for this device.

**string** `broadcast` - Broadcast address for device.

**array** `ipv6` - List of IPv6 addresses

**array**

- **struct ipv6** address
  - **string** `address` - IPv6 address of this network device
  - **string** `netmask` - IPv6 netmask of this network device
  - **string** `scope` - IPv6 address scope

**struct dmi\_info** The server's dmi info

**struct DMI**

- **string** `vendor`
- **string** `system`
- **string** `product`

- `string asset`
- `string board`
- `string bios_release` - (optional)
- `string bios_vendor` - (optional)
- `string bios_version` - (optional)

struct `cpu_info` The server's cpu info

struct `CPU`

- `string cache`
- `string family`
- `string mhz`
- `string flags`
- `string model`
- `string vendor`
- `string arch`
- `string stepping`
- `string count`
- `int socket_count` (if available)"

array `subscribed_channels` - List of subscribed channels

array

- struct `channel`
  - `int channel_id` - The channel id.
  - `string channel_label` - The channel label.

array `active_guest_system_ids` - List of virtual guest system ids for active guests

array

- `int guest_id` - The guest's system id.

## 34.49. METHOD: LISTADMINISTRATORS

Returns a list of users which can administer the system.

### Parameters

`string usernameKey`



- `string sessionKey`
- `int serverId`

### Returns

- `array`
  - `struct user`
    - `int id`
    - `string login`
    - `string login_uc` - upper case version of the login
    - `boolean enabled` - true if user is enabled, false if the user is disabled

## 34.50. METHOD: LISTALLINSTALLABLEPACKAGES

Get the list of all installable packages for a given system.

### Parameters

- `string sessionKey`
- `int serverId`

### Returns

- `struct package`
  - `string name`
  - `string version`
  - `string release`
  - `string epoch`
  - `int id`
  - `string arch_label`

## 34.51. METHOD: LISTBASECHANNELS

Returns a list of subscribable base channels.

Deprecated - being replaced by `listSubscribableBaseChannels(string sessionKey, int serverId)`

### Parameters

- `string sessionKey`
- `int serverId`

## Returns

- array
  - struct `channel`
    - int `id` - Base Channel ID.
    - string `name` - Name of channel.
    - string `label` - Label of Channel
    - int `current_base` - 1 indicates it is the current base channel

## 34.52. METHOD: LISTCHILDCHANNELS

Returns a list of subscribable child channels. This only shows channels the system is *\*not\** currently subscribed to.

Deprecated - being replaced by `listSubscribableChildChannels(string sessionKey, int serverId)`

## Parameters

- string `sessionKey`
- int `serverId`

## Returns

- array
  - struct `child channel`
    - int `id`
    - string `name`
    - string `label`
    - string `summary`
    - string `has_license`
    - string `gpg_key_url`

## 34.53. METHOD: LISTDUPLICATESBYHOSTNAME

List duplicate systems by Hostname.

## Parameters

- string `sessionKey`

## Returns

- array

- struct **Duplicate Group**
  - string **hostname**
  - array **systems**
    - struct **system**
      - int **systemId**
      - string **systemName**
      - dateTime.iso8601 **last\_checkin** - Last time server successfully checked in

### 34.54. METHOD: LISTDUPLICATESBYIP

List duplicate systems by IP Address.

#### Parameters

- string **sessionKey**

#### Returns

- array
  - struct **Duplicate Group**
    - string **ip**
    - array **systems**
      - struct **system**
        - int **systemId**
        - string **systemName**
        - dateTime.iso8601 **last\_checkin** - Last time server successfully checked in

### 34.55. METHOD: LISTDUPLICATESBYMAC

List duplicate systems by Mac Address.

#### Parameters

- string **sessionKey**

#### Returns

- array
  - struct **Duplicate Group**
    - string **mac**

- **array systems**
  - **struct system**
    - **int systemId**
    - **string systemName**
    - **dateTime.iso8601 last\_checkin** - Last time server successfully checked in

## 34.56. METHOD: LISTELIGIBLEFLEXGUESTS

List eligible flex guests accessible to the user

### Parameters

- **string sessionKey**

### Returns

**array**

**struct channel** family system group

**int id**

**string label**

**string name**

**array**

- **int** - systems

## 34.57. METHOD: LISTEXTRAPACKAGES

List extra packages for a system

### Parameters

- **string sessionKey**
- **int serverId**

### Returns

- **array**
  - **struct package**
    - **string name**
    - **string version**
    - **string release**

- string `epoch` - returned only if non-zero
- string `arch`
- date "installtime" - returned only if known

## 34.58. METHOD: LISTFLEXGUESTS

List flex guests accessible to the user

### Parameters

- string `sessionKey`

### Returns

array

struct `channel` family system group

int `id`

string `label`

string `name`

array

- int - systems

## 34.59. METHOD: LISTGROUPS

List the available groups for a given system.

### Parameters

- string `sessionKey`
- int `serverId`

### Returns

- array
  - struct `system` group
    - int `id` - server group id
    - int `subscribed` - 1 if the given server is subscribed to this server group, 0 otherwise
    - string `system_group_name` - Name of the server group
    - string `sgid` - server group id (Deprecated)

## 34.60. METHOD: LISTINACTIVESYSTEMS

Lists systems that have been inactive for the default period of inactivity

### Parameters

- string `sessionKey`

### Returns

- array
  - struct `system`
    - int `id`
    - string `name`
    - `dateTime.iso8601 last_checkin` - Last time server successfully checked in

## 34.61. METHOD: LISTINACTIVESYSTEMS

Lists systems that have been inactive for the specified number of days..

### Parameters

- string `sessionKey`
- int `days`

### Returns

- array
  - struct `system`
    - int `id`
    - string `name`
    - `dateTime.iso8601 last_checkin` - Last time server successfully checked in

## 34.62. METHOD: LISTLATESTAVAILABLEPACKAGE

Get the latest available version of a package for each system

### Parameters

- string `sessionKey`
- array
  - int - `serverId`
- string `packageName`

### Returns

array

**struct system**

**int id** - server ID

**string name** - server name

**struct package** package structure

**struct package**

- **int id**
- **string name**
- **string version**
- **string release**
- **string epoch**
- **string arch**

### 34.63. METHOD: LISTLATESTINSTALLABLEPACKAGES

Get the list of latest installable packages for a given system.

#### Parameters

- **string sessionKey**
- **int serverId**

#### Returns

- **array**
  - **struct package**
    - **string name**
    - **string version**
    - **string release**
    - **string epoch**
    - **int id**
    - **string arch\_label**

### 34.64. METHOD: LISTLATESTUPGRADABLEPACKAGES

Get the list of latest upgradable packages for a given system.

#### Parameters

- `string sessionKey`
- `int serverId`

## Returns

- `array`
  - `struct package`
    - `string name`
    - `string arch`
    - `string from_version`
    - `string from_release`
    - `string from_epoch`
    - `string to_version`
    - `string to_release`
    - `string to_epoch`
    - `string to_package_id`

## 34.65. METHOD: LISTNEWERINSTALLEDPACKAGES

Given a package name, version, release, and epoch, returns the list of packages installed on the system w/ the same name that are newer.

## Parameters

- `string sessionKey`
- `int serverId`
- `string name` - Package name.
- `string version` - Package version.
- `string release` - Package release.
- `string epoch` - Package epoch.

## Returns

- `array`
  - `struct package`
    - `string name`
    - `string version`



- `string release`
- `string epoch`

## 34.66. METHOD: LISTNOTES

Provides a list of notes associated with a system.

### Parameters

- `string sessionKey`
- `int serverId`

### Returns

- `array`
  - `struct note details`
    - `int id`
    - `string subject` - Subject of the note
    - `string note` - Contents of the note
    - `int system_id` - The id of the system associated with the note
    - `string creator` - Creator of the note if exists (optional)
    - `date "updated"` - Date of the last note update

## 34.67. METHOD: LISTOLDERINSTALLEDPACKAGES

Given a package name, version, release, and epoch, returns the list of packages installed on the system with the same name that are older.

### Parameters

- `string sessionKey`
- `int serverId`
- `string name` - Package name.
- `string version` - Package version.
- `string release` - Package release.
- `string epoch` - Package epoch.

### Returns

- `array`
  - `struct package`

- `string name`
- `string version`
- `string release`
- `string epoch`

## 34.68. METHOD: LISTOUTOFDATESYSTEMS

Returns list of systems needing package updates.

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct system`
    - `int id`
    - `string name`
    - `dateTime.iso8601 last_checkin` - Last time server successfully checked in

## 34.69. METHOD: LISTPACKAGEPROFILES

List the package profiles in this organization

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct package profile`
    - `int id`
    - `string name`
    - `string channel`

## 34.70. METHOD: LISTPACKAGES

List the installed packages for a given system. The attribute `installtime` is returned since API version 10.10.

### Parameters

- `string sessionKey`
- `int serverId`

## Returns

- `array`
  - `struct package`
    - `string name`
    - `string version`
    - `string release`
    - `string epoch`
    - `string arch`
    - `date "installtime"` - returned only if known

## 34.71. METHOD: LISTPACKAGESFROMCHANNEL

Provides a list of packages installed on a system that are also contained in the given channel. The installed package list did not include arch information before RHEL 5, so it is arch unaware. RHEL 5 systems do upload the arch information, and thus are arch aware.

## Parameters

- `string sessionKey`
- `int serverId`
- `string channelLabel`

## Returns

- `array`
  - `struct package`
    - `string name`
    - `string version`
    - `string release`
    - `string epoch`
    - `int id`
    - `string arch_label`
    - `string path` - The path on that file system that the package resides

- string **provider** - The provider of the package, determined by the gpg key it was signed with.
- dateTime.iso8601 **last\_modified**

## 34.72. METHOD: LISTPHYSICALSYSTEMS

Returns a list of all Physical servers visible to the user.

### Parameters

- string **sessionKey**

### Returns

- array
  - struct **system**
    - int **id**
    - string **name**
    - dateTime.iso8601 **last\_checkin** - Last time server successfully checked in

## 34.73. METHOD: LISTSUBSCRIBABLEBASECHANNELS

Returns a list of subscribable base channels.

### Parameters

- string **sessionKey**
- int **serverId**

### Returns

- array
  - struct **channel**
    - int **id** - Base Channel ID.
    - string **name** - Name of channel.
    - string **label** - Label of Channel
    - int **current\_base** - 1 indicates it is the current base channel

## 34.74. METHOD: LISTSUBSCRIBABLECHILDCHANNELS

Returns a list of subscribable child channels. This only shows channels the system is *\*not\** currently subscribed to.

### Parameters

- `string sessionKey`
- `int serverId`

## Returns

- `array`
  - `struct child channel`
    - `int id`
    - `string name`
    - `string label`
    - `string summary`
    - `string has_license`
    - `string gpg_key_url`

## 34.75. METHOD: LISTSUBSCRIBEDCHILDCHANNELS

Returns a list of subscribed child channels.

## Parameters

- `string sessionKey`
- `int serverId`

## Returns

`array`

`struct channel`

`int id`

`string name`

`string label`

`string arch_name`

`string arch_label`

`string summary`

`string description`

`string checksum_label`

`dateTime.iso8601 last_modified`

`string maintainer_name`

string **maintainer\_email**

string **maintainer\_phone**

string **support\_policy**

string **gpg\_key\_url**

string **gpg\_key\_id**

string **gpg\_key\_fp**

dateTime.iso8601 **yumrepo\_last\_sync** - (optional)

string **end\_of\_life**

string **parent\_channel\_label**

string **clone\_original**

array

- struct **contentSources**
  - int **id**
  - string **label**
  - string **sourceUrl**
  - string **type**

## 34.76. METHOD: LISTSYSTEMEVENTS

List system events of the specified type for given server. "actionType" should be exactly the string returned in the `action_type` field from the `listSystemEvents(sessionKey, serverId)` method. For example, 'Package Install' or 'Initiate a kickstart for a virtual guest.'

### Parameters

- string **sessionKey**
- int **serverId** - ID of system.
- string **actionType** - Type of the action.

### Returns

- array
  - struct **action**
    - int **failed\_count** - Number of times action failed.
    - string **modified** - Date modified. (Deprecated by `modified_date`)

- **dateTime.iso8601 modified\_date** - Date modified.
- **string created** - Date created. (Deprecated by **created\_date**)
- **dateTime.iso8601 created\_date** - Date created.
- **string action\_type**
- **int successful\_count** - Number of times action was successful.
- **string earliest\_action** - Earliest date this action will occur.
- **int archived** - If this action is archived. (1 or 0)
- **string scheduler\_user** - available only if concrete user has scheduled the action
- **string prerequisite** - Pre-requisite action. (optional)
- **string name** - Name of this action.
- **int id** - Id of this action.
- **string version** - Version of action.
- **string completion\_time** - The date/time the event was completed. Format ->YYYY-MM-dd hh:mm:ss.ms Eg ->2007-06-04 13:58:13.0. (optional) (Deprecated by **completed\_date**)
- **dateTime.iso8601 completed\_date** - The date/time the event was completed. (optional)
- **string pickup\_time** - The date/time the action was picked up. Format ->YYYY-MM-dd hh:mm:ss.ms Eg ->2007-06-04 13:58:13.0. (optional) (Deprecated by **pickup\_date**)
- **dateTime.iso8601 pickup\_date** - The date/time the action was picked up. (optional)
- **string result\_msg** - The result string after the action executes at the client machine. (optional)
- **array additional\_info** - This array contains additional information for the event, if available.
  - **struct info**
    - **string detail** - The detail provided depends on the specific event. For example, for a package event, this will be the package name, for an errata event, this will be the advisory name and synopsis, for a config file event, this will be path and optional revision information...etc.
    - **string result** - The result (if included) depends on the specific event. For example, for a package or errata event, no result is included, for a config file event, the result might include an error (if one occurred, such as the file was missing) or in the case of a config file comparison it might include the differences found.

## 34.77. METHOD: LISTSYSTEMEVENTS

List all system events for given server. This includes *\*all\** events for the server since it was registered. This may require the caller to filter the results to fetch the specific events they are looking for.

### Parameters

- string **sessionKey**
- int **serverId** - ID of system.

### Returns

- array
  - struct **action**
    - int **failed\_count** - Number of times action failed.
    - string **modified** - Date modified. (Deprecated by **modified\_date**)
    - dateTime.iso8601 **modified\_date** - Date modified.
    - string **created** - Date created. (Deprecated by **created\_date**)
    - dateTime.iso8601 **created\_date** - Date created.
    - string **action\_type**
    - int **successful\_count** - Number of times action was successful.
    - string **earliest\_action** - Earliest date this action will occur.
    - int **archived** - If this action is archived. (1 or 0)
    - string **scheduler\_user** - available only if concrete user has scheduled the action
    - string **prerequisite** - Pre-requisite action. (optional)
    - string **name** - Name of this action.
    - int **id** - Id of this action.
    - string **version** - Version of action.
    - string **completion\_time** - The date/time the event was completed. Format ->YYYY-MM-dd hh:mm:ss.ms Eg ->2007-06-04 13:58:13.0. (optional) (Deprecated by **completed\_date**)
    - dateTime.iso8601 **completed\_date** - The date/time the event was completed. (optional)
    - string **pickup\_time** - The date/time the action was picked up. Format ->YYYY-MM-dd hh:mm:ss.ms Eg ->2007-06-04 13:58:13.0. (optional) (Deprecated by **pickup\_date**)
    - dateTime.iso8601 **pickup\_date** - The date/time the action was picked up. (optional)



- string **result\_msg** - The result string after the action executes at the client machine. (optional)
- array **additional\_info** - This array contains additional information for the event, if available.
  - struct **info**
    - string **detail** - The detail provided depends on the specific event. For example, for a package event, this will be the package name, for an errata event, this will be the advisory name and synopsis, for a config file event, this will be path and optional revision information...etc.
    - string **result** - The result (if included) depends on the specific event. For example, for a package or errata event, no result is included, for a config file event, the result might include an error (if one occurred, such as the file was missing) or in the case of a config file comparison it might include the differences found.

Available since: 10.8

## 34.78. METHOD: LISTSYSTEMS

Returns a list of all servers visible to the user.

### Parameters

- string **sessionKey**

### Returns

- array
  - struct **system**
    - int **id**
    - string **name**
    - dateTime.iso8601 **last\_checkin** - Last time server successfully checked in

## 34.79. METHOD: LISTSYSTEMSWITHEXTRAPACKAGES

List systems with extra packages

### Parameters

- string **sessionKey**

### Returns

- array
  - int **id** - System ID
  - string **name** - System profile name

- `int extra_pkg_count` - Extra packages count

## 34.80. METHOD: LISTSYSTEMSWITHPACKAGE

Lists the systems that have the given installed package

### Parameters

- `string sessionKey`
- `int pid` - the package id

### Returns

- array
  - `struct system`
    - `int id`
    - `string name`
    - `dateTime.iso8601 last_checkin` - Last time server successfully checked in

## 34.81. METHOD: LISTSYSTEMSWITHPACKAGE

Lists the systems that have the given installed package

### Parameters

- `string sessionKey`
- `string name` - the package name
- `string version` - the package version
- `string release` - the package release

### Returns

- array
  - `struct system`
    - `int id`
    - `string name`
    - `dateTime.iso8601 last_checkin` - Last time server successfully checked in

## 34.82. METHOD: LISTUNGROUPEDSYSTEMS

List systems that are not associated with any system groups.

### Parameters

- `string sessionKey`

## Returns

- `array`
  - `struct system`
    - `int id` - server id
    - `string name`

## 34.83. METHOD: LISTUSERSYSTEMS

List systems for a given user.

### Parameters

- `string sessionKey`
- `string login` - User's login name.

## Returns

- `array`
  - `struct system`
    - `int id`
    - `string name`
    - `dateTime.iso8601 last_checkin` - Last time server successfully checked in

## 34.84. METHOD: LISTUSERSYSTEMS

List systems for the logged in user.

### Parameters

- `string sessionKey`

## Returns

- `array`
  - `struct system`
    - `int id`
    - `string name`
    - `dateTime.iso8601 last_checkin` - Last time server successfully checked in

## 34.85. METHOD: LISTVIRTUALGUESTS

Lists the virtual guests for agiven virtual host

### Parameters

- string `sessionKey`
- int `sid` - the virtual host's id

### Returns

- array
  - struct `virtual system`
    - int `id`
    - string `name`
    - string `guest_name` - The virtual guest name as provided by the virtual host
    - `dateTime.iso8601 last_checkin` - Last time server successfully checked in.
    - string `uuid`

## 34.86. METHOD: LISTVIRTUALHOSTS

Lists the virtual hosts visible to the user

### Parameters

- string `sessionKey`

### Returns

- array
  - struct `system`
    - int `id`
    - string `name`
    - `dateTime.iso8601 last_checkin` - Last time server successfully checked in

## 34.87. METHOD: OBTAINREACTIVATIONKEY

Obtains a reactivation key for this server.

### Parameters

- string `sessionKey`
- int `serverId`

### Returns

- `string`

### 34.88. METHOD: OBTAINREACTIVATIONKEY

Obtains a reactivation key for this server.

#### Parameters

- `string systemid` - systemid file

#### Returns

- `string`

Available since: 10.10

### 34.89. METHOD: PROVISIONSYSTEM

Provision a system using the specified kickstart profile.

#### Parameters

- `string sessionKey`
- `int serverId` - ID of the system to be provisioned.
- `string profileName` - Kickstart profile to use.

#### Returns

- `int` - ID of the action scheduled, otherwise exception thrown on error

### 34.90. METHOD: PROVISIONSYSTEM

Provision a system using the specified kickstart profile.

#### Parameters

- `string sessionKey`
- `int serverId` - ID of the system to be provisioned.
- `string profileName` - Kickstart profile to use.
- `dateTime.iso8601 earliestDate`

#### Returns

- `int` - ID of the action scheduled, otherwise exception thrown on error

### 34.91. METHOD: PROVISIONVIRTUALGUEST

Provision a guest on the host specified. Defaults to: memory=512MB, vcpu=1, storage=3GB, mac\_address=random.

## Parameters

- string `sessionKey`
- int `serverId` - ID of host to provision guest on.
- string `guestName`
- string `profileName` - Kickstart profile to use.

## Returns

- int - 1 on success, exception thrown otherwise.

## 34.92. METHOD: PROVISIONVIRTUALGUEST

Provision a guest on the host specified. This schedules the guest for creation and will begin the provisioning process when the host checks in or if OSAD is enabled will begin immediately. Defaults to `mac_address=random`.

## Parameters

- string `sessionKey`
- int `serverId` - ID of host to provision guest on.
- string `guestName`
- string `profileName` - Kickstart Profile to use.
- int `memoryMb` - Memory to allocate to the guest
- int `vcpus` - Number of virtual CPUs to allocate to the guest.
- int `storageGb` - Size of the guests disk image.

## Returns

- int - 1 on success, exception thrown otherwise.

## 34.93. METHOD: PROVISIONVIRTUALGUEST

Provision a guest on the host specified. This schedules the guest for creation and will begin the provisioning process when the host checks in or if OSAD is enabled will begin immediately.

## Parameters

- string `sessionKey`
- int `serverId` - ID of host to provision guest on.
- string `guestName`
- string `profileName` - Kickstart Profile to use.
- int `memoryMb` - Memory to allocate to the guest

- `int vcpus` - Number of virtual CPUs to allocate to the guest.
- `int storageGb` - Size of the guests disk image.
- `string macAddress` - `macAddress` to give the guest's virtual networking hardware.

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.94. METHOD: REMOVEENTITLEMENTS

Remove addon entitlements from a server. Entitlements a server does not have are quietly ignored.

### Parameters

- `string sessionKey`
- `int serverId`
- `array`
  - `string` - `entitlement_label`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.95. METHOD: SCHEDULEAPPLYERRATA

Schedules an action to apply errata updates to multiple systems.

### Parameters

- `string sessionKey`
- `array`
  - `int` - `serverId`
- `array`
  - `int` - `errataId`

### Returns

- `array`
  - `int` - `actionId`

Available since: 13.0

## 34.96. METHOD: SCHEDULEAPPLYERRATA

Schedules an action to apply errata updates to multiple systems at a given date/time.

### Parameters

- string `sessionKey`
- array
  - int - `serverId`
- array
  - int - `errataId`
- `dateTime.iso8601 earliestOccurrence`

### Returns

- array
  - int - `actionId`

Available since: 13.0

## 34.97. METHOD: SCHEDULEAPPLYERRATA

Schedules an action to apply errata updates to a system.

### Parameters

- string `sessionKey`
- int `serverId`
- array
  - int - `errataId`

### Returns

- array
  - int - `actionId`

Available since: 13.0

## 34.98. METHOD: SCHEDULEAPPLYERRATA

Schedules an action to apply errata updates to a system at a given date/time.

### Parameters

- string `sessionKey`
- int `serverId`
- array



- int - errataId
- dateTime.iso8601 earliestOccurrence

### Returns

- array
  - int - actionId

Available since: 13.0

## 34.99. METHOD: SCHEDULECERTIFICATEUPDATE

Schedule update of client certificate

### Parameters

- string sessionKey
- int serverId

### Returns

- int actionId - The action id of the scheduled action

## 34.100. METHOD: SCHEDULECERTIFICATEUPDATE

Schedule update of client certificate at given date and time

### Parameters

- string sessionKey
- int serverId
- dateTime.iso860 date

### Returns

- int actionId - The action id of the scheduled action

## 34.101. METHOD: SCHEDULEGUESTACTION

Schedules a guest action for the specified virtual guest for a given date/time.

### Parameters

- string sessionKey
- int sid - the system Id of the guest
- string state - One of the following actions 'start', 'suspend', 'resume', 'restart', 'shutdown'.
- dateTime.iso8601 date - the time/date to schedule the action

## Returns

- `int actionId` - The action id of the scheduled action

### 34.102. METHOD: SCHEDULEGUESTACTION

Schedules a guest action for the specified virtual guest for the current time.

## Parameters

- `string sessionKey`
- `int sid` - the system Id of the guest
- `string state` - One of the following actions 'start', 'suspend', 'resume', 'restart', 'shutdown'.

## Returns

- `int actionId` - The action id of the scheduled action

### 34.103. METHOD: SCHEDULEHARDWAREREFRESH

Schedule a hardware refresh for a system.

## Parameters

- `string sessionKey`
- `int serverId`
- `dateTime.iso8601 earliestOccurrence`

## Returns

- `int actionId` - The action id of the scheduled action

Available since: 13.0

### 34.104. METHOD: SCHEDULEPACKAGEINSTALL

Schedule package installation for a system.

## Parameters

- `string sessionKey`
- `array`
  - `int` - `serverId`
- `array`
  - `int` `packageId`
- `dateTime.iso8601 earliestOccurrence`

## Returns

- array
  - int - actionId

## 34.105. METHOD: SCHEDULEPACKAGEINSTALL

Schedule package installation for a system.

### Parameters

- string `sessionKey`
- int `serverId`
- array
  - int `packageId`
- `dateTime.iso8601 earliestOccurrence`

## Returns

- int `actionId` - The action id of the scheduled action

Available since: 13.0

## 34.106. METHOD: SCHEDULEPACKAGEREFRESH

Schedule a package list refresh for a system.

### Parameters

- string `sessionKey`
- int `serverId`
- `dateTime.iso8601 earliestOccurrence`

## Returns

- int - ID of the action scheduled, otherwise exception thrown on error

## 34.107. METHOD: SCHEDULEPACKAGEREMOVE

Schedule package removal for a system.

### Parameters

- string `sessionKey`
- int `serverId`
- array

- `int packageId`
- `dateTime.iso8601 earliestOccurrence`

### Returns

- `int` - ID of the action scheduled, otherwise exception thrown on error

## 34.108. METHOD: SCHEDULEREBOOT

Schedule a reboot for a system.

### Parameters

- `string sessionKey`
- `int serverId`
- `dateTime.iso860 earliestOccurrence`

### Returns

- `int actionId` - The action id of the scheduled action

Available since: 13.0

## 34.109. METHOD: SCHEDULESCRIPTRUN

Schedule a script to run.

### Parameters

- `string sessionKey`
- `array`
  - `int` - System IDs of the servers to run the script on.
- `string username` - User to run script as.
- `string groupname` - Group to run script as.
- `int timeout` - Seconds to allow the script to run before timing out.
- `string script` - Contents of the script to run.
- `dateTime.iso8601 earliestOccurrence` - Earliest the script can run.

### Returns

- `int` - ID of the script run action created. Can be used to fetch results with `system.getScriptResults`.

## 34.110. METHOD: SCHEDULESCRIPTRUN

Schedule a script to run.

## Parameters

- `string sessionKey`
- `int serverId` - ID of the server to run the script on.
- `string username` - User to run script as.
- `string groupname` - Group to run script as.
- `int timeout` - Seconds to allow the script to run before timing out.
- `string script` - Contents of the script to run.
- `dateTime.iso8601 earliestOccurrence` - Earliest the script can run.

## Returns

- `int` - ID of the script run action created. Can be used to fetch results with `system.getScriptResults`.

## 34.111. METHOD: SCHEDULESYNCPACKAGESWITHSYSTEM

Sync packages from a source system to a target.

## Parameters

- `string sessionKey`
- `int targetServerId` - Target system to apply package changes to.
- `int sourceServerId` - Source system to retrieve package state from.
- `array`
  - `int packageId` - Package IDs to be synced.
- `dateTime.iso8601 date` - Date to schedule action for

## Returns

- `int actionId` - The action id of the scheduled action

Available since: 13.0

## 34.112. METHOD: SEARCHBYNAME

Returns a list of system IDs whose name matches the supplied regular expression(defined by

## Parameters

- `string sessionKey`
- `string regexp` - A regular expression

## Returns

- array
  - struct `system`
    - `int id`
    - `string name`
    - `dateTime.iso8601 last_checkin` - Last time server successfully checked in

### 34.113. METHOD: SETBASECHANNEL

Assigns the server to a new `baseChannel`.

Deprecated - being replaced by `system.setBaseChannel(string sessionKey, int serverId, string channelLabel)`

## Parameters

- `string sessionKey`
- `int serverId`
- `int channelId`

## Returns

- `int` - 1 on success, exception thrown otherwise.

### 34.114. METHOD: SETBASECHANNEL

Assigns the server to a new base channel. If the user provides an empty string for the `channelLabel`, the current base channel and all child channels will be removed from the system.

## Parameters

- `string sessionKey`
- `int serverId`
- `string channelLabel`

## Returns

- `int` - 1 on success, exception thrown otherwise.

### 34.115. METHOD: SETCHILDCHANNELS

Subscribe the given server to the child channels provided. This method will unsubscribe the server from any child channels that the server is currently subscribed to, but that are not included in the list. The user may provide either a list of channel ids (`int`) or a list of channel labels (`string`) as input.

## Parameters

- `string sessionKey`
- `int serverId`
- `array`
  - `int (deprecated)` or `string - channelId (deprecated)` or `channelLabel`

## Returns

- `int - 1` on success, exception thrown otherwise.

## 34.116. METHOD: SETCUSTOMVALUES

Set custom values for the specified server.

### Parameters

- `string sessionKey`
- `int serverId`
- `struct Map` of custom labels to custom values
  - `string custom info label"`
  - `string value`

## Returns

- `int - 1` on success, exception thrown otherwise.

## 34.117. METHOD: SETDETAILS

Set server details. All arguments are optional and will only be modified if included in the struct.

### Parameters

- `string sessionKey`
- `int serverId` - ID of server to lookup details for.
- `struct server` details
  - `string profile_name` - System's profile name
  - `string base_entitlement` - System's base entitlement label. (`enterprise_entitled` or `sw_mgr_entitled`)
  - `boolean auto_errata_update` - True if system has auto errata updates enabled
  - `string description` - System description
  - `string address1` - System's address line 1.
  - `string address2` - System's address line 2.

- string `city`
- string `state`
- string `country`
- string `building`
- string `room`
- string `rack`

### Returns

- int - 1 on success, exception thrown otherwise.

## 34.118. METHOD: SETGROUPMEMBERSHIP

Set a servers membership in a given group.

### Parameters

- string `sessionKey`
- int `serverId`
- int `serverGroupId`
- boolean `member` - '1' to assign the given server to the given server group, '0' to remove the given server from the given server group.

### Returns

- int - 1 on success, exception thrown otherwise.

## 34.119. METHOD: SETGUESTCPUS

Schedule an action of a guest's host, to set that guest's CPU allocation

### Parameters

- string `sessionKey`
- int `sid` - The guest's system id
- int `numOfCpus` - The number of virtual cpus to allocate to the guest

### Returns

- int `actionID` - the action Id for the schedule action on the host system.

## 34.120. METHOD: SETGUESTMEMORY

Schedule an action of a guest's host, to set that guest's memory allocation



## Parameters

- `string sessionKey`
- `int sid` - The guest's system id
- `int memory` - The amount of memory to allocate to the guest

## Returns

- `int actionID` - the action Id for the schedule action on the host system.

## 34.121. METHOD: SETLOCKSTATUS

Set server lock status.

## Parameters

- `string sessionKey`
- `int serverId`
- `boolean lockStatus` - true to lock the system, false to unlock the system.

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.122. METHOD: SETPRIMARYINTERFACE

Sets new primary network interface

## Parameters

- `string sessionKey`
- `int serverId`
- `string interfaceName`

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.123. METHOD: SETPROFILENAME

Set the profile name for the server.

## Parameters

- `string sessionKey`
- `int serverId`
- `string name` - Name of the profile.

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.124. METHOD: SETVARIABLES

Sets a list of kickstart variables in the cobbler system record for the specified server. Note: This call assumes that a system record exists in cobbler for the given system and will raise an XMLRPC fault if that is not the case. To create a system record over xmlrpc use `system.createSystemRecord` To create a system record in the Web UI please go to System - Specified System - Provisioning - Select a Kickstart profile - Create Cobbler System Record.

## Parameters

- `string sessionKey`
- `int serverId`
- `boolean netboot`
- `array`
  - `struct kickstart variable`
    - `string key`
    - `string or int "value"`

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.125. METHOD: TAGLATESTSNAPSHOT

Tags latest system snapshot

## Parameters

- `string sessionKey`
- `int serverId`
- `string tagName`

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.126. METHOD: UNENTITLE

Unentitle the system completely

## Parameters

- `string systemid` - systemid file

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.127. METHOD: UPGRADEENTITLEMENT

Adds an entitlement to a given server.

### Parameters

- `string sessionKey`
- `int serverId`
- `string entitlementName` - One of: 'enterprise\_entitled', 'provisioning\_entitled', 'monitoring\_entitled', 'nonlinux\_entitled', 'virtualization\_host', or 'virtualization\_host\_platform'.

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 34.128. METHOD: WHOREGISTERED

Returns information about the user who registered the system

### Parameters

- `string sessionKey`
- `int sid` - Id of the system in question

## Returns

- `struct user`
  - `int id`
  - `string login`
  - `string login_uc` - upper case version of the login
  - `boolean enabled` - true if user is enabled, false if the user is disabled

## CHAPTER 35. NAMESPACE: SYSTEM.CONFIG

Provides methods to access and modify many aspects of configuration channels and server association. basically system.config name space

### 35.1. METHOD: ADDCHANNELS

Given a list of servers and configuration channels, this method appends the configuration channels to either the top or the bottom (whichever you specify) of a system's subscribed configuration channels list. The ordering of the configuration channels provided in the add list is maintained while adding. If one of the configuration channels in the 'add' list has been previously subscribed by a server, the subscribed channel will be re-ranked to the appropriate place.

#### Parameters

- string **sessionKey**
- array
  - int - IDs of the systems to add the channels to.
- array
  - string - List of configuration channel labels in the ranked order.
- boolean **addToTop**
  - true - to prepend the given channels list to the top of the configuration channels list of a server
  - false - to append the given channels list to the bottom of the configuration channels list of a server

#### Returns

- int - 1 on success, exception thrown otherwise.

### 35.2. METHOD: CREATEORUPDATEPATH

Create a new file (text or binary) or directory with the given path, or update an existing path on a server.

#### Parameters

- string **sessionKey**
- int **serverId**
- string **path** - the configuration file/directory path
- boolean **isDir**
  - True - if the path is a directory
  - False - if the path is a file

- **struct path info**
  - **string contents** - Contents of the file (text or base64 encoded if binary) ((only for non-directories)
  - **boolean contents\_enc64** - Identifies base64 encoded content (default: disabled, only for non-directories).
  - **string owner** - Owner of the file/directory.
  - **string group** - Group name of the file/directory.
  - **string permissions** - Octal file/directory permissions (eg: 644)
  - **string macro-start-delimiter** - Config file macro end delimiter. Use null or empty string to accept the default. (only for non-directories)
  - **string macro-end-delimiter** - Config file macro end delimiter. Use null or empty string to accept the default. (only for non-directories)
  - **string selinux\_ctx** - SeLinux context (optional)
  - **int revision** - next revision number, auto increment for null
  - **boolean binary** - mark the binary content, if True, base64 encoded content is expected (only for non-directories)
- **int commitToLocal**
  - 1 - to commit configuration files to the system's local override configuration channel
  - 0 - to commit configuration files to the system's sandbox configuration channel

## Returns

**struct Configuration** Revision information

**string type**

- file
- directory
- symlink

**string path** - File Path

**string target\_path** - Symbolic link Target File Path. Present for Symbolic links only.

**string channel** - Channel Name

**string contents** - File contents (base64 encoded according to the contents\_enc64 attribute)

**boolean contents\_enc64** - Identifies base64 encoded content

**int revision** - File Revision

`dateTime.iso8601 creation` - Creation Date

`dateTime.iso8601 modified` - Last Modified Date

`string owner` - File Owner. Present for files or directories only.

`string group` - File Group. Present for files or directories only.

`int permissions` - File Permissions (Deprecated). Present for files or directories only.

`string permissions_mode` - File Permissions. Present for files or directories only.

`string selinux_ctx` - SELinux Context (optional).

`boolean binary` - true/false , Present for files only.

`string md5` - File's md5 signature. Present for files only.

`string macro-start-delimiter` - Macro start delimiter for a config file. Present for text files only.

`string macro-end-delimiter` - Macro end delimiter for a config file. Present for text files only.

Available since: 10.2

### 35.3. METHOD: CREATEORUPDATESYMLINK

Create a new symbolic link with the given path, or update an existing path.

#### Parameters

- `string sessionKey`
- `int serverId`
- `string path` - the configuration file/directory path
- `struct path info`
  - `string target_path` - The target path for the symbolic link
  - `string selinux_ctx` - SELinux Security context (optional)
  - `int revision` - next revision number, auto increment for null
- `int commitToLocal`
  - 1 - to commit configuration files to the system's local override configuration channel
  - 0 - to commit configuration files to the system's sandbox configuration channel

#### Returns

`struct Configuration` Revision information

`string type`

- `file`

- `directory`
- `symlink`

`string path` - File Path

`string target_path` - Symbolic link Target File Path. Present for Symbolic links only.

`string channel` - Channel Name

`string contents` - File contents (base64 encoded according to the `contents_enc64` attribute)

`boolean contents_enc64` - Identifies base64 encoded content

`int revision` - File Revision

`dateTime.iso8601 creation` - Creation Date

`dateTime.iso8601 modified` - Last Modified Date

`string owner` - File Owner. Present for files or directories only.

`string group` - File Group. Present for files or directories only.

`int permissions` - File Permissions (Deprecated). Present for files or directories only.

`string permissions_mode` - File Permissions. Present for files or directories only.

`string selinux_ctx` - SELinux Context (optional).

`boolean binary` - true/false , Present for files only.

`string md5` - File's md5 signature. Present for files only.

`string macro-start-delimiter` - Macro start delimiter for a config file. Present for text files only.

`string macro-end-delimiter` - Macro end delimiter for a config file. Present for text files only.

Available since: 10.2

## 35.4. METHOD: DELETEFILES

Removes file paths from a local or sandbox channel of a server.

### Parameters

- `string sessionKey`
- `int serverId`
- `array`
  - `string` - paths to remove.
- `boolean deleteFromLocal`

- `True` - to delete configuration file paths from the system's local override configuration channel
- `False` - to delete configuration file paths from the system's sandbox configuration channel

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 35.5. METHOD: DEPLOYALL

Schedules a deploy action for all the configuration files on the given list of systems.

### Parameters

- `string sessionKey`
- `array`
  - `int` - id of the systems to schedule configuration files deployment
- `dateTime.iso8601 date` - Earliest date for the deploy action.

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 35.6. METHOD: LISTCHANNELS

List all global configuration channels associated to a system in the order of their ranking.

### Parameters

- `string sessionKey`
- `int serverId`

## Returns

`array`

`struct Configuration` Channel information

`int id`

`int orgId`

`string label`

`string name`

`string description`

`struct configChannelType`

`struct Configuration` Channel Type information



- `int id`
- `string label`
- `string name`
- `int priority`

## 35.7. METHOD: LISTFILES

Return the list of files in a given channel.

### Parameters

- `string sessionKey`
- `int serverId`
- `int listLocal`
  - 1 - to return configuration files in the system's local override configuration channel
  - 0 - to return configuration files in the system's sandbox configuration channel

### Returns

array

`struct Configuration` File information

`string type`

- `file`
- `directory`
- `symlink`

`string path` - File Path

`string channel_label` - the label of the central configuration channel that has this file. Note this entry only shows up if the file has not been overridden by a central channel.

`struct channel_type`

`struct Configuration` Channel Type information

- `int id`
- `string label`
- `string name`
- `int priority`

`dateTime.iso8601 last_modified` - Last Modified Date

## 35.8. METHOD: LOOKUPFILEINFO

Given a list of paths and a server, returns details about the latest revisions of the paths.

### Parameters

- `string sessionKey`
- `int serverId`
- `array`
  - `string` - paths to lookup on.
- `int searchLocal`
  - 1 - to search configuration file paths in the system's local override configuration or systems subscribed central channels
  - 0 - to search configuration file paths in the system's sandbox configuration channel

### Returns

`array`

`struct Configuration` Revision information

`string type`

- `file`
- `directory`
- `symlink`

`string path` - File Path

`string target_path` - Symbolic link Target File Path. Present for Symbolic links only.

`string channel` - Channel Name

`string contents` - File contents (base64 encoded according to the `contents_enc64` attribute)

`boolean contents_enc64` - Identifies base64 encoded content

`int revision` - File Revision

`dateTime.iso8601 creation` - Creation Date

`dateTime.iso8601 modified` - Last Modified Date

`string owner` - File Owner. Present for files or directories only.

`string group` - File Group. Present for files or directories only.

`int permissions` - File Permissions (Deprecated). Present for files or directories only.

`string permissions_mode` - File Permissions. Present for files or directories only.

string **selinux\_ctx** - SELinux Context (optional).

boolean **binary** - true/false , Present for files only.

string **md5** - File's md5 signature. Present for files only.

string **macro-start-delimiter** - Macro start delimiter for a config file. Present for text files only.

string **macro-end-delimiter** - Macro end delimiter for a config file. Present for text files only.

Available since: 10.2

## 35.9. METHOD: REMOVECHANNELS

Remove config channels from the given servers.

### Parameters

- string **sessionKey**
- array
  - int - the IDs of the systems from which you would like to remove configuration channels..
- array
  - string - List of configuration channel labels to remove.

### Returns

- int - 1 on success, exception thrown otherwise.

## 35.10. METHOD: SETCHANNELS

Replace the existing set of config channels on the given servers. Channels are ranked according to their order in the configChannelLabels array.

### Parameters

- string **sessionKey**
- array
  - int - IDs of the systems to set the channels on.
- array
  - string - List of configuration channel labels in the ranked order.

### Returns

- int - 1 on success, exception thrown otherwise.

## CHAPTER 36. NAMESPACE: SYSTEM.CRASH

Provides methods to access and modify software crash information.

### 36.1. METHOD: CREATECRASHNOTE

Create a crash note

#### Parameters

- `string sessionKey`
- `int crashId`
- `string subject`
- `string details`

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 36.2. METHOD: DELETECRASH

Delete a crash with given crash id.

#### Parameters

- `string sessionKey`
- `int crashId`

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 36.3. METHOD: DELETECRASHNOTE

Delete a crash note

#### Parameters

- `string sessionKey`
- `int crashNoteId`

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 36.4. METHOD: GETCRASHCOUNTINFO

Return date of last software crashes report for given system

### Parameters

- `string sessionKey`
- `int serverId`

### Returns

- `struct Crash Count Information`
  - `int total_count` - Total number of software crashes for a system
  - `int unique_count` - Number of unique software crashes for a system
  - `dateTime.iso8601 last_report` - Date of the last software crash report

## 36.5. METHOD: GETCRASHFILE

Download a crash file.

### Parameters

- `string sessionKey`
- `int crashFileId`

### Returns

- `base64` - base64 encoded crash file.

## 36.6. METHOD: GETCRASHFILEURL

Get a crash file download url.

### Parameters

- `string sessionKey`
- `int crashFileId`

### Returns

- `string` - The crash file download url

## 36.7. METHOD: GETCRASHNOTESFORCRASH

List crash notes for crash

### Parameters

- `string sessionKey`
- `int crashId`

### Returns

- array
  - struct `crashNote`
    - int `id`
    - string `subject`
    - string `details`
    - string `updated`

## 36.8. METHOD: GETCRASHOVERVIEW

Get Software Crash Overview

### Parameters

- string `sessionKey`

### Returns

- array
  - struct `crash`
    - string `uuid` - Crash UUID
    - string `component` - Package component (set if unique and non empty)
    - int `crash_count` - Number of crashes occurred
    - int `system_count` - Number of systems affected
    - `dateTime.iso8601 last_report` - Last crash occurrence

## 36.9. METHOD: GETCRASHESBYUUID

List software crashes with given UUID

### Parameters

- string `sessionKey`
- string `uuid`

### Returns

- array
  - struct `crash`
    - int `server_id` - ID of the server the crash occurred on
    - string `server_name` - Name of the server the crash occurred on

- `int crash_id` - ID of the crash with given UUID
- `int crash_count` - Number of times the crash with given UUID occurred
- `string crash_component` - Crash component
- `dateTime.iso8601 last_report` - Last crash occurrence

## 36.10. METHOD: LISTSYSTEMCRASHFILES

Return list of crash files for given crash id.

### Parameters

- `string sessionKey`
- `int crashId`

### Returns

- `array`
  - `struct crashFile`
    - `int id`
    - `string filename`
    - `string path`
    - `int filesize`
    - `boolean is_uploaded`
    - `date "created"`
    - `date "modified"`

## 36.11. METHOD: LISTSYSTEMCRASHES

Return list of software crashes for a system.

### Parameters

- `string sessionKey`
- `int serverId`

### Returns

- `array`
  - `struct crash`
    - `int id`

- **string crash**
- **string path**
- **int count**
- **string uuid**
- **string analyzer**
- **string architecture**
- **string cmdline**
- **string component**
- **string executable**
- **string kernel**
- **string reason**
- **string username**
- **date "created"**
- **date "modified"**



## CHAPTER 37. NAMESPACE: SYSTEM.CUSTOMINFO

Provides methods to access and modify custom system information.

### 37.1. METHOD: CREATEKEY

Create a new custom key

#### Parameters

- `string sessionKey`
- `string keyLabel` - new key's label
- `string keyDescription` - new key's description

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 37.2. METHOD: DELETEKEY

Delete an existing custom key and all systems' values for the key.

#### Parameters

- `string sessionKey`
- `string keyLabel` - new key's label

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 37.3. METHOD: LISTALLKEYS

List the custom information keys defined for the user's organization.

#### Parameters

- `string sessionKey`

#### Returns

- `array`
  - `struct custom info`
    - `int id`
    - `string label`
    - `string description`
    - `int system_count`

■ `dateTime.iso8601 last_modified`

## 37.4. METHOD: UPDATEKEY

Update description of a custom key

### Parameters

- `string sessionKey`
- `string keyLabel` - key to change
- `string keyDescription` - new key's description

### Returns

- `int` - 1 on success, exception thrown otherwise.

## CHAPTER 38. NAMESPACE: SYSTEM.PROVISIONING.SNAPSHOT

Provides methods to access and delete system snapshots.

### 38.1. METHOD: ADDTAGTOSNAPSHOT

Adds tag to snapshot

#### Parameters

- `string sessionKey`
- `int snapshotId` - Id of the snapshot
- `string tag` - Name of the snapshot tag

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 38.2. METHOD: DELETESNAPSHOT

Deletes a snapshot with the given snapshot id

#### Parameters

- `string sessionKey`
- `int snapshotId` - Id of snapshot to delete

#### Returns

- `int` - 1 on success, exception thrown otherwise.

Available since: 10.1

### 38.3. METHOD: DELETESNAPSHOTS

Deletes all snapshots across multiple systems based on the given date criteria. For example,

- If the user provides `startDate` only, all snapshots created either on or after the date provided will be removed.
- If user provides `startDate` and `endDate`, all snapshots created on or between the dates provided will be removed.
- If the user doesn't provide a `startDate` and `endDate`, all snapshots will be removed.

#### Parameters

- `string sessionKey`
- `struct date details`

- `dateTime.iso8601 startDate` - Optional, unless `endDate` is provided.
- `dateTime.iso8601 endDate` - Optional.

## Returns

- `int` - 1 on success, exception thrown otherwise.

Available since: 10.1

## 38.4. METHOD: DELETESNAPSHOTS

Deletes all snapshots for a given system based on the date criteria. For example,

- If the user provides `startDate` only, all snapshots created either on or after the date provided will be removed.
- If user provides `startDate` and `endDate`, all snapshots created on or between the dates provided will be removed.
- If the user doesn't provide a `startDate` and `endDate`, all snapshots associated with the server will be removed.

## Parameters

- `string sessionKey`
- `int sid` - system id of system to delete snapshots for
- `struct date details`
  - `dateTime.iso8601 startDate` - Optional, unless `endDate` is provided.
  - `dateTime.iso8601 endDate` - Optional.

## Returns

- `int` - 1 on success, exception thrown otherwise.

Available since: 10.1

## 38.5. METHOD: LISTSNAPSHOTCONFIGFILES

List the config files associated with a snapshot.

## Parameters

- `string sessionKey`
- `int snapId`

## Returns

`array`

`struct Configuration` Revision information

string **type**

- file
- directory
- symlink

string **path** - File Path

string **target\_path** - Symbolic link Target File Path. Present for Symbolic links only.

string **channel** - Channel Name

string **contents** - File contents (base64 encoded according to the **contents\_enc64** attribute)

boolean **contents\_enc64** - Identifies base64 encoded content

int **revision** - File Revision

dateTime.iso8601 **creation** - Creation Date

dateTime.iso8601 **modified** - Last Modified Date

string **owner** - File Owner. Present for files or directories only.

string **group** - File Group. Present for files or directories only.

int **permissions** - File Permissions (Deprecated). Present for files or directories only.

string **permissions\_mode** - File Permissions. Present for files or directories only.

string **selinux\_ctx** - SELinux Context (optional).

boolean **binary** - true/false , Present for files only.

string **md5** - File's md5 signature. Present for files only.

string **macro-start-delimiter** - Macro start delimiter for a config file. Present for text files only.

string **macro-end-delimiter** - Macro end delimiter for a config file. Present for text files only.

Available since: 10.2

## 38.6. METHOD: LISTSNAPSHOTPACKAGES

List the packages associated with a snapshot.

### Parameters

- string **sessionKey**
- int **snapId**

### Returns

- array
  - struct **package nvera**
    - string **name**
    - string **epoch**
    - string **version**
    - string **release**
    - string **arch**

Available since: 10.1

## 38.7. METHOD: LISTSNAPSHOTS

List snapshots for a given system. A user may optionally provide a start and end date to narrow the snapshots that will be listed. For example,

- If the user provides `startDate` only, all snapshots created either on or after the date provided will be returned.
- If user provides `startDate` and `endDate`, all snapshots created on or between the dates provided will be returned.
- If the user doesn't provide a `startDate` and `endDate`, all snapshots associated with the server will be returned.

### Parameters

- string **sessionKey**
- int **serverId**
- struct **date details**
  - `dateTime.iso8601 startDate` - Optional, unless `endDate` is provided.
  - `dateTime.iso8601 endDate` - Optional.

### Returns

- array
  - struct **server snapshot**
    - int **id**
    - string **reason** - the reason for the snapshot's existence
    - `dateTime.iso8601 created`
    - array **channels**

- string `labels` of channels associated with the snapshot
- array `groups`
  - string `Names` of server groups associated with the snapshot
- array `entitlements`
  - string `Names` of system entitlements associated with the snapshot
- array `config_channels`
  - string `Labels` of config channels the snapshot is associated with.
- array `tags`
  - string `Tag` names associated with this snapshot.
- string `Invalid_reason` - If the snapshot is invalid, this is the reason (optional).

Available since: 10.1

## 38.8. METHOD: ROLLBACKTOSNAPSHOT

Rollbacks server to snapshot

### Parameters

- string `sessionKey`
- int `serverId`
- int `snapshotId` - Id of the snapshot

### Returns

- int - 1 on success, exception thrown otherwise.

## 38.9. METHOD: ROLLBACKTOTAG

Rollbacks server to snapshot

### Parameters

- string `sessionKey`
- int `serverId`
- string `tagName` - Name of the snapshot tag

### Returns

- int - 1 on success, exception thrown otherwise.

## 38.10. METHOD: ROLLBACKTOTAG

Rollbacks server to snapshot

### Parameters

- string `sessionKey`
- string `tagName` - Name of the snapshot tag

### Returns

- int - 1 on success, exception thrown otherwise.



## CHAPTER 39. NAMESPACE: SYSTEM.SCAP

Provides methods to schedule SCAP scans and access the results.

### 39.1. METHOD: DELETEXCCDFSCAN

Delete OpenSCAP XCCDF Scan from Spacewalk database. Note that only those SCAP Scans can be deleted which have passed their retention period.

#### Parameters

- string `sessionKey`
- int `Id` of XCCDF scan (`xid`).

#### Returns

- boolean - indicates success of the operation.

### 39.2. METHOD: GETXCCDFSCANDetails

Get details of given OpenSCAP XCCDF scan.

#### Parameters

- string `sessionKey`
- int `Id` of XCCDF scan (`xid`).

#### Returns

- struct **OpenSCAP XCCDF Scan**
  - int `xid` - XCCDF TestResult id
  - int `sid` - serverId
  - int `action_id` - Id of the parent action.
  - string `path` - Path to XCCDF document
  - string `oscap_parameters` - oscap command-line arguments.
  - string `test_result` - Identifier of XCCDF TestResult.
  - string `benchmark` - Identifier of XCCDF Benchmark.
  - string `benchmark_version` - Version of the Benchmark.
  - string `profile` - Identifier of XCCDF Profile.
  - string `profile_title` - Title of XCCDF Profile.
  - dateTime.iso8601 `start_time` - Client machine time of scan start.

- `dateTime.iso8601 end_time` - Client machine time of scan completion.
- `string errors` - Stderr output of scan.
- `bool "deletable"` - Indicates whether the scan can be deleted.

### 39.3. METHOD: GETXCCDFSCANRULERESULTS

Return a full list of RuleResults for given OpenSCAP XCCDF scan.

#### Parameters

- `string sessionKey`
- `int Id` of XCCDF scan (`xid`).

#### Returns

- `array`
  - `struct OpenSCAP XCCDF RuleResult`
    - `string idref` - idref from XCCDF document.
    - `string result` - Result of evaluation.
    - `string idents` - Comma separated list of XCCDF idents.

### 39.4. METHOD: LISTXCCDFSCANS

Return a list of finished OpenSCAP scans for a given system.

#### Parameters

- `string sessionKey`
- `int serverId`

#### Returns

- `array`
  - `struct OpenSCAP XCCDF Scan`
    - `int xid` - XCCDF TestResult ID
    - `string profile` - XCCDF Profile
    - `string path` - Path to XCCDF document
    - `dateTime.iso8601 completed` - Scan completion time

### 39.5. METHOD: SCHEDULEXCCDFSCAN

Schedule OpenSCAP scan.

## Parameters

- string `sessionKey`
- array
  - int - `serverId`
- string `Path` to xccdf content on targeted systems.
- string `Additional` parameters for oscap tool.

## Returns

- int - ID if SCAP action created.

## 39.6. METHOD: SCHEDULEXCCDFSCAN

Schedule OpenSCAP scan.

### Parameters

- string `sessionKey`
- array
  - int - `serverId`
- string `Path` to xccdf content on targeted systems.
- string `Additional` parameters for oscap tool.
- `dateTime.iso8601 date` - The date to schedule the action

### Returns

- int - ID if SCAP action created.

## 39.7. METHOD: SCHEDULEXCCDFSCAN

Schedule Scap XCCDF scan.

### Parameters

- string `sessionKey`
- int `serverId`
- string `Path` to xccdf content on targeted system.
- string `Additional` parameters for oscap tool.

### Returns

- int - ID of the scap action created.

## 39.8. METHOD: SCHEDULEXCCDFSCAN

Schedule Scap XCCDF scan.

### Parameters

- string **sessionKey**
- int **serverId**
- string **Path** to xccdf content on targeted system.
- string **Additional** parameters for oscap tool.
- dateTime.iso8601 **date** - The date to schedule the action

### Returns

- int - ID of the scap action created.

## CHAPTER 40. NAMESPACE: SYSTEM.SEARCH

Provides methods to perform system search requests using the search server.

### 40.1. METHOD: DEVIDEDESCRIPTION

List the systems which match the device description.

#### Parameters

- string `sessionKey`
- string `searchTerm`

#### Returns

- array
  - struct `system`
    - int `id`
    - string `name`
    - dateTime.iso8601 `last_checkin` - Last time server successfully checked in
    - string `hostname`
    - string `ip`
    - string `hw_description` - hw description if not null
    - string `hw_device_id` - hw device id if not null
    - string `hw_vendor_id` - hw vendor id if not null
    - string `hw_driver` - hw driver if not null

### 40.2. METHOD: DEVIDEDRIVER

List the systems which match this device driver.

#### Parameters

- string `sessionKey`
- string `searchTerm`

#### Returns

- array
  - struct `system`
    - int `id`

- `string name`
- `dateTime.iso8601 last_checkin` - Last time server successfully checked in
- `string hostname`
- `string ip`
- `string hw_description` - hw description if not null
- `string hw_device_id` - hw device id if not null
- `string hw_vendor_id` - hw vendor id if not null
- `string hw_driver` - hw driver if not null

### 40.3. METHOD: DEVICEID

List the systems which match this device id

#### Parameters

- `string sessionKey`
- `string searchTerm`

#### Returns

- `array`
  - `struct system`
    - `int id`
    - `string name`
    - `dateTime.iso8601 last_checkin` - Last time server successfully checked in
    - `string hostname`
    - `string ip`
    - `string hw_description` - hw description if not null
    - `string hw_device_id` - hw device id if not null
    - `string hw_vendor_id` - hw vendor id if not null
    - `string hw_driver` - hw driver if not null

### 40.4. METHOD: DEVICEVENDORID

List the systems which match this device vendor\_id

#### Parameters

- string `sessionKey`
- string `searchTerm`

## Returns

- array
  - struct `system`
    - int `id`
    - string `name`
    - dateTime.iso8601 `last_checkin` - Last time server successfully checked in
    - string `hostname`
    - string `ip`
    - string `hw_description` - hw description if not null
    - string `hw_device_id` - hw device id if not null
    - string `hw_vendor_id` - hw vendor id if not null
    - string `hw_driver` - hw driver if not null

## 40.5. METHOD: HOSTNAME

List the systems which match this hostname

### Parameters

- string `sessionKey`
- string `searchTerm`

## Returns

- array
  - struct `system`
    - int `id`
    - string `name`
    - dateTime.iso8601 `last_checkin` - Last time server successfully checked in
    - string `hostname`
    - string `ip`
    - string `hw_description` - hw description if not null

- string `hw_device_id` - hw device id if not null
- string `hw_vendor_id` - hw vendor id if not null
- string `hw_driver` - hw driver if not null

## 40.6. METHOD: IP

List the systems which match this ip.

### Parameters

- string `sessionKey`
- string `searchTerm`

### Returns

- array
  - struct `system`
    - int `id`
    - string `name`
    - dateTime.iso8601 `last_checkin` - Last time server successfully checked in
    - string `hostname`
    - string `ip`
    - string `hw_description` - hw description if not null
    - string `hw_device_id` - hw device id if not null
    - string `hw_vendor_id` - hw vendor id if not null
    - string `hw_driver` - hw driver if not null

## 40.7. METHOD: NAMEANDDESCRIPTION

List the systems which match this name or description

### Parameters

- string `sessionKey`
- string `searchTerm`

### Returns

- array
  - struct `system`



- `int id`
- `string name`
- `dateTime.iso8601 last_checkin` - Last time server successfully checked in
- `string hostname`
- `string ip`
- `string hw_description` - hw description if not null
- `string hw_device_id` - hw device id if not null
- `string hw_vendor_id` - hw vendor id if not null
- `string hw_driver` - hw driver if not null

## 40.8. METHOD: UUID

List the systems which match this UUID

### Parameters

- `string sessionKey`
- `string searchTerm`

### Returns

- `array`
  - `struct system`
    - `int id`
    - `string name`
    - `dateTime.iso8601 last_checkin` - Last time server successfully checked in
    - `string hostname`
    - `string ip`
    - `string hw_description` - hw description if not null
    - `string hw_device_id` - hw device id if not null
    - `string hw_vendor_id` - hw vendor id if not null
    - `string hw_driver` - hw driver if not null

## CHAPTER 41. NAMESPACE: SYSTEMGROUP

Provides methods to access and modify system groups.

### 41.1. METHOD: ADDORREMOVEADMINS

Add or remove administrators to/from the given group. Satellite and Organization administrators are granted access to groups within their organization by default; therefore, users with those roles should not be included in the array provided. Caller must be an organization administrator.

#### Parameters

- `string sessionKey`
- `string systemGroupName`
- `array`
  - `string` - `loginName` - User's `loginName`
- `int add` - 1 to add administrators, 0 to remove.

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 41.2. METHOD: ADDORREMOVESYSTEMS

Add/remove the given servers to a system group.

#### Parameters

- `string sessionKey`
- `string systemGroupName`
- `array`
  - `int` - `serverId`
- `boolean add` - True to add to the group, False to remove.

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 41.3. METHOD: CREATE

Create a new system group.

#### Parameters

- `string sessionKey`
- `string name` - Name of the system group.

- string **description** - Description of the system group.

## Returns

- struct **Server Group**
  - int **id**
  - string **name**
  - string **description**
  - int **org\_id**
  - int **system\_count**

## 41.4. METHOD: DELETE

Delete a system group.

### Parameters

- string **sessionKey**
- string **systemGroupName**

### Returns

- int - 1 on success, exception thrown otherwise.

## 41.5. METHOD: GETDETAILS

Retrieve details of a ServerGroup based on it's id

### Parameters

- string **sessionKey**
- int **systemGroupId**

### Returns

- struct **Server Group**
  - int **id**
  - string **name**
  - string **description**
  - int **org\_id**
  - int **system\_count**

## 41.6. METHOD: GETDETAILS

Retrieve details of a ServerGroup based on it's name

### Parameters

- string `sessionKey`
- string `systemGroupName`

### Returns

- struct `Server Group`
  - int `id`
  - string `name`
  - string `description`
  - int `org_id`
  - int `system_count`

## 41.7. METHOD: LISTACTIVESYSTEMSINGROUP

Lists active systems within a server group

### Parameters

- string `sessionKey`
- string `systemGroupName`

### Returns

- array
  - int - `server_id`

## 41.8. METHOD: LISTADMINISTRATORS

Returns the list of users who can administer the given group. Caller must be a system group admin or an organization administrator.

### Parameters

- string `sessionKey`
- string `systemGroupName`

### Returns

- array
  - struct `user`
    - int `id`

- `string login`
- `string login_uc` - upper case version of the login
- `boolean enabled` - true if user is enabled, false if the user is disabled

## 41.9. METHOD: LISTALLGROUPS

Retrieve a list of system groups that are accessible by the logged in user.

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct Server Group`
    - `int id`
    - `string name`
    - `string description`
    - `int org_id`
    - `int system_count`

## 41.10. METHOD: LISTGROUPSWITHNOASSOCIATEDADMINS

Returns a list of system groups that do not have an administrator. (who is not an organization administrator, as they have implicit access to system groups) Caller must be an organization administrator.

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct Server Group`
    - `int id`
    - `string name`
    - `string description`
    - `int org_id`
    - `int system_count`

## 41.11. METHOD: LISTINACTIVESYSTEMSINGROUP

Lists inactive systems within a server group using a specified inactivity time.

### Parameters

- string `sessionKey`
- string `systemGroupName`
- int `daysInactive` - Number of days a system must not check in to be considered inactive.

### Returns

- array
  - int - `server_id`

## 41.12. METHOD: LISTINACTIVESYSTEMSINGROUP

Lists inactive systems within a server group using the default 1 day threshold.

### Parameters

- string `sessionKey`
- string `systemGroupName`

### Returns

- array
  - int - `server_id`

## 41.13. METHOD: LISTSYSTEMS

Return a list of systems associated with this system group. User must have access to this system group.

### Parameters

- string `sessionKey`
- string `systemGroupName`

### Returns

- array
  - struct `server` details
    - int `id` - System id
    - string `profile_name`

- **string base\_entitlement** - System's base entitlement label. (enterprise\_entitled or sw\_mgr\_entitled)
- **array string**
  - **addon\_entitlements** System's addon entitlements labels, including monitoring\_entitled, provisioning\_entitled, virtualization\_host, virtualization\_host\_platform
- **boolean auto\_update** - True if system has auto errata updates enabled.
- **string release** - The Operating System release (i.e. 4AS, 5Server)
- **string address1**
- **string address2**
- **string city**
- **string state**
- **string country**
- **string building**
- **string room**
- **string rack**
- **string description**
- **string hostname**
- **dateTime.iso8601 last\_boot**
- **string osa\_status** - Either 'unknown', 'offline', or 'online'.
- **boolean lock\_status** - True indicates that the system is locked. False indicates that the system is unlocked.
- **string virtualization** - Virtualization type - for virtual guests only (optional)

## 41.14. METHOD: LISTSYSTEMSMINIMAL

Return a list of systems associated with this system group. User must have access to this system group.

### Parameters

- **string sessionKey**
- **string systemGroupName**

### Returns

- **array**

- `struct system`
  - `int id`
  - `string name`
  - `dateTime.iso8601 last_checkin` - Last time server successfully checked in

## 41.15. METHOD: SCHEDULEAPPLYERRATATOACTIVE

Schedules an action to apply errata updates to active systems from a group.

### Parameters

- `string sessionKey`
- `string systemGroupName`
- `array`
  - `int - errataId`

### Returns

- `array`
  - `int - actionId`

Available since: 13.0

## 41.16. METHOD: SCHEDULEAPPLYERRATATOACTIVE

Schedules an action to apply errata updates to active systems from a group at a given date/time.

### Parameters

- `string sessionKey`
- `string systemGroupName`
- `array`
  - `int - errataId`
- `dateTime.iso8601 earliestOccurrence`

### Returns

- `array`
  - `int - actionId`

Available since: 13.0

## 41.17. METHOD: UPDATE



Update an existing system group.

### Parameters

- `string sessionKey`
- `string systemGroupName`
- `string description`

### Returns

- `struct Server Group`
  - `int id`
  - `string name`
  - `string description`
  - `int org_id`
  - `int system_count`

## CHAPTER 42. NAMESPACE: USER.EXTERNAL

If you are using IPA integration to allow authentication of users from an external IPA server (rare) the users will still need to be created in the Satellite database. Methods in this namespace allow you to configure some specifics of how this happens, like what organization they are created in or what roles they will have. These options can also be set in the web admin interface.

### 42.1. METHOD: CREATEEXTERNALGROUPTOROLEMAP

Externally authenticated users may be members of external groups. You can use these groups to assign additional roles to the users when they log in. Can only be called by a `satellite_admin`.

#### Parameters

- `string sessionKey`
- `string name` - Name of the external group. Must be unique.
- `array`
  - `string` - `role` - Can be any of: `satellite_admin`, `org_admin` (implies all other roles except for `satellite_admin`), `channel_admin`, `config_admin`, `system_group_admin`, `activation_key_admin`, or `monitoring_admin`.

#### Returns

- `struct externalGroup`
  - `string name`
  - `array roles`
    - `string role`

### 42.2. METHOD: CREATEEXTERNALGROUPTOSYSTEMGROUPMAP

Externally authenticated users may be members of external groups. You can use these groups to give access to server groups to the users when they log in. Can only be called by an `org_admin`.

#### Parameters

- `string sessionKey`
- `string name` - Name of the external group. Must be unique.
- `array`
  - `string` - `groupName` - The names of the server groups to grant access to.

#### Returns

- `struct externalGroup`
  - `string name`

- array roles
  - string role

### 42.3. METHOD: DELETEEXTERNALGROUPTOROLEMAP

Delete the role map for an external group. Can only be called by a satellite\_admin.

#### Parameters

- string sessionKey
- string name - Name of the external group.

#### Returns

- int - 1 on success, exception thrown otherwise.

### 42.4. METHOD: DELETEEXTERNALGROUPTOSYSTEMGROUPMAP

Delete the server group map for an external group. Can only be called by an org\_admin.

#### Parameters

- string sessionKey
- string name - Name of the external group.

#### Returns

- int - 1 on success, exception thrown otherwise.

### 42.5. METHOD: GETDEFAULTORG

Get the default org that users should be added in if orgunit from IPA server isn't found or is disabled. Can only be called by a satellite\_admin.

#### Parameters

- string sessionKey

#### Returns

- int - Id of the default organization. 0 if there is no default.

### 42.6. METHOD: GETEXTERNALGROUPTOROLEMAP

Get a representation of the role mapping for an external group. Can only be called by a satellite\_admin.

#### Parameters

- string sessionKey
- string name - Name of the external group.

## Returns

- `struct externalGroup`
  - `string name`
  - `array roles`
    - `string role`

## 42.7. METHOD: GETEXTERNALGROUPTOSYSTEMGROUPMAP

Get a representation of the server group mapping for an external group. Can only be called by an `org_admin`.

### Parameters

- `string sessionKey`
- `string name` - Name of the external group.

### Returns

- `struct externalGroup`
  - `string name`
  - `array roles`
    - `string role`

## 42.8. METHOD: GETKEEPTEMPORARYROLES

Get whether we should keep roles assigned to users because of their IPA groups even after they log in through a non-IPA method. Can only be called by a `satellite_admin`.

### Parameters

- `string sessionKey`

### Returns

- `boolean` - True if we should keep roles after users log in through non-IPA method, false otherwise.

## 42.9. METHOD: GETUSEORGUNIT

Get whether we place users into the organization that corresponds to the "orgunit" set on the IPA server. The orgunit name must match exactly the Satellite organization name. Can only be called by a `satellite_admin`.

### Parameters

- `string sessionKey`

## Returns

- boolean - True if we should use the IPA orgunit to determine which organization to create the user in, false otherwise.

## 42.10. METHOD: LISTEXTERNALGROUPTOROLEMAPS

List role mappings for all known external groups. Can only be called by a `satellite_admin`.

### Parameters

- string `sessionKey`

### Returns

- array
  - struct `externalGroup`
    - string `name`
    - array `roles`
      - string `role`

## 42.11. METHOD: LISTEXTERNALGROUPTOSYSTEMGROUPMAPS

List server group mappings for all known external groups. Can only be called by an `org_admin`.

### Parameters

- string `sessionKey`

### Returns

- array
  - struct `externalGroup`
    - string `name`
    - array `roles`
      - string `role`

## 42.12. METHOD: SETDEFAULTORG

Set the default org that users should be added in if orgunit from IPA server isn't found or is disabled. Can only be called by a `satellite_admin`.

### Parameters

- string `sessionKey`

- `int defaultOrg` - Id of the organization to set as the default org. 0 if there should not be a default organization.

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 42.13. METHOD: SETEXTERNALGROUPROLES

Update the roles for an external group. Replace previously set roles with the ones passed in here. Can only be called by a `satellite_admin`.

### Parameters

- `string sessionKey`
- `string name` - Name of the external group.
- `array`
  - `string` - `role` - Can be any of: `satellite_admin`, `org_admin` (implies all other roles except for `satellite_admin`), `channel_admin`, `config_admin`, `system_group_admin`, `activation_key_admin`, or `monitoring_admin`.

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 42.14. METHOD: SETEXTERNALGROUPSYSTEMGROUPS

Update the server groups for an external group. Replace previously set server groups with the ones passed in here. Can only be called by an `org_admin`.

### Parameters

- `string sessionKey`
- `string name` - Name of the external group.
- `array`
  - `string` - `groupName` - The names of the server groups to grant access to.

## Returns

- `int` - 1 on success, exception thrown otherwise.

## 42.15. METHOD: SETKEEPTEMPORARYROLES

Set whether we should keep roles assigned to users because of their IPA groups even after they log in through a non-IPA method. Can only be called by a `satellite_admin`.

### Parameters

- `string sessionKey`

- boolean **keepRoles** - True if we should keep roles after users log in through non-IPA method, false otherwise.

## Returns

- int - 1 on success, exception thrown otherwise.

## 42.16. METHOD: SETUSEORGUNIT

Set whether we place users into the organization that corresponds to the "orgunit" set on the IPA server. The orgunit name must match exactly the Satellite organization name. Can only be called by a `satellite_admin`.

## Parameters

- string **sessionKey**
- boolean **useOrgUnit** - True if we should use the IPA orgunit to determine which organization to create the user in, false otherwise.

## Returns

- int - 1 on success, exception thrown otherwise.

## CHAPTER 43. NAMESPACE: USER

User namespace contains methods to access common user functions available from the web user interface.

### 43.1. METHOD: ADDASSIGNEDSYSTEMGROUP

Add system group to user's list of assigned system groups.

#### Parameters

- `string sessionKey`
- `string login` - User's login name.
- `string serverGroupName`
- `boolean setDefault` - Should system group also be added to user's list of default system groups.

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 43.2. METHOD: ADDASSIGNEDSYSTEMGROUPS

Add system groups to user's list of assigned system groups.

#### Parameters

- `string sessionKey`
- `string login` - User's login name.
- `array`
  - `string` - `serverGroupName`
- `boolean setDefault` - Should system groups also be added to user's list of default system groups.

#### Returns

- `int` - 1 on success, exception thrown otherwise.

### 43.3. METHOD: ADDDEFAULTSYSTEMGROUP

Add system group to user's list of default system groups.

#### Parameters

- `string sessionKey`
- `string login` - User's login name.



- `string serverGroupName`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 43.4. METHOD: ADDDEFAULTSYSTEMGROUPS

Add system groups to user's list of default system groups.

### Parameters

- `string sessionKey`
- `string login` - User's login name.
- `array`
  - `string` - `serverGroupName`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 43.5. METHOD: ADDROLE

Adds a role to a user.

### Parameters

- `string sessionKey`
- `string login` - User login name to update.
- `string role` - Role label to add. Can be any of: `satellite_admin`, `org_admin`, `channel_admin`, `config_admin`, `system_group_admin`, `activation_key_admin`, or `monitoring_admin`.

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 43.6. METHOD: CREATE

Create a new user.

### Parameters

- `string sessionKey`
- `string desiredLogin` - Desired login name, will fail if already in use.
- `string desiredPassword`
- `string firstName`

- string `lastName`
- string `email` - User's e-mail address.

### Returns

- int - 1 on success, exception thrown otherwise.

## 43.7. METHOD: CREATE

Create a new user.

### Parameters

- string `sessionKey`
- string `desiredLogin` - Desired login name, will fail if already in use.
- string `desiredPassword`
- string `firstName`
- string `lastName`
- string `email` - User's e-mail address.
- int `usePamAuth` - 1 if you wish to use PAM authentication for this user, 0 otherwise.

### Returns

- int - 1 on success, exception thrown otherwise.

## 43.8. METHOD: DELETE

Delete a user.

### Parameters

- string `sessionKey`
- string `login` - User login name to delete.

### Returns

- int - 1 on success, exception thrown otherwise.

## 43.9. METHOD: DISABLE

Disable a user.

### Parameters

- string `sessionKey`
- string `login` - User login name to disable.

## Returns

- int - 1 on success, exception thrown otherwise.

## 43.10. METHOD: ENABLE

Enable a user.

### Parameters

- string `sessionKey`
- string `login` - User login name to enable.

## Returns

- int - 1 on success, exception thrown otherwise.

## 43.11. METHOD: GETCREATEDDEFAULTSYSTEMGROUP

Returns the current value of the `CreateDefaultSystemGroup` setting. If `True` this will cause there to be a system group created (with the same name as the user) every time a new user is created, with the user automatically given permission to that system group and the system group being set as the default group for the user (so every time the user registers a system it will be placed in that system group by default). This can be useful if different users will administer different groups of servers in the same organization. Can only be called by an `org_admin`.

### Parameters

- string `sessionKey`

## Returns

- int - 1 on success, exception thrown otherwise.

## 43.12. METHOD: GETDETAILS

Returns the details about a given user.

### Parameters

- string `sessionKey`
- string `login` - User's login name.

## Returns

- struct `user details`
  - string `first_names` - deprecated, use `first_name`
  - string `first_name`
  - string `last_name`

- string **email**
- int **org\_id**
- string **org\_name**
- string **prefix**
- string **last\_login\_date**
- string **created\_date**
- boolean **enabled** - true if user is enabled, false if the user is disabled
- boolean **use\_pam** - true if user is configured to use PAM authentication
- boolean **read\_only** - true if user is readonly
- boolean **errata\_notification** - true if errata e-mail notification is enabled for the user

### 43.13. METHOD: GETLOGGEDINTIME

Returns the time user last logged in.

#### Parameters

- string **sessionKey**
- string **login** - User's login name.

#### Returns

- **dateTime.iso8601**

### 43.14. METHOD: LISTASSIGNABLEROLES

Returns a list of user roles that this user can assign to others.

#### Parameters

- string **sessionKey**

#### Returns

- array
  - string - (role label)

### 43.15. METHOD: LISTASSIGNEDSYSTEMGROUPS

Returns the system groups that a user can administer.

#### Parameters

- string **sessionKey**

- `string login` - User's login name.

## Returns

- array
  - struct `system group`
    - `int id`
    - `string name`
    - `string description`
    - `int system_count`
    - `int org_id` - Organization ID for this system group.

## 43.16. METHOD: LISTDEFAULTSYSTEMGROUPS

Returns a user's list of default system groups.

### Parameters

- `string sessionKey`
- `string login` - User's login name.

## Returns

- array
  - struct `system group`
    - `int id`
    - `string name`
    - `string description`
    - `int system_count`
    - `int org_id` - Organization ID for this system group.

## 43.17. METHOD: LISTROLES

Returns a list of the user's roles.

### Parameters

- `string sessionKey`
- `string login` - User's login name.

## Returns

- `array`
  - `string` - (role label)

## 43.18. METHOD: LISTUSERS

Returns a list of users in your organization.

### Parameters

- `string sessionKey`

### Returns

- `array`
  - `struct user`
    - `int id`
    - `string login`
    - `string login_uc` - upper case version of the login
    - `boolean enabled` - true if user is enabled, false if the user is disabled

## 43.19. METHOD: REMOVEASSIGNEDSYSTEMGROUP

Remove system group from the user's list of assigned system groups.

### Parameters

- `string sessionKey`
- `string login` - User's login name.
- `string serverGroupName`
- `boolean setDefault` - Should system group also be removed from the user's list of default system groups.

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 43.20. METHOD: REMOVEASSIGNEDSYSTEMGROUPS

Remove system groups from a user's list of assigned system groups.

### Parameters

- `string sessionKey`
- `string login` - User's login name.

- `array`
  - `string` - `serverGroupName`
- `boolean setDefault` - Should system groups also be removed from the user's list of default system groups.

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 43.21. METHOD: REMOVEDEFAULTSYSTEMGROUP

Remove a system group from user's list of default system groups.

### Parameters

- `string sessionKey`
- `string login` - User's login name.
- `string serverGroupName`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 43.22. METHOD: REMOVEDEFAULTSYSTEMGROUPS

Remove system groups from a user's list of default system groups.

### Parameters

- `string sessionKey`
- `string login` - User's login name.
- `array`
  - `string` - `serverGroupName`

### Returns

- `int` - 1 on success, exception thrown otherwise.

## 43.23. METHOD: REMOVEROLE

Remove a role from a user.

### Parameters

- `string sessionKey`
- `string login` - User login name to update.

- string **role** - Role label to remove. Can be any of: `satellite_admin`, `org_admin`, `channel_admin`, `config_admin`, `system_group_admin`, `activation_key_admin`, or `monitoring_admin`.

## Returns

- int - 1 on success, exception thrown otherwise.

## 43.24. METHOD: SETCREATEDDEFAULTSYSTEMGROUP

Sets the value of the `CreateDefaultSystemGroup` setting. If `True` this will cause there to be a system group created (with the same name as the user) every time a new user is created, with the user automatically given permission to that system group and the system group being set as the default group for the user (so every time the user registers a system it will be placed in that system group by default). This can be useful if different users will administer different groups of servers in the same organization. Can only be called by an `org_admin`.

## Parameters

- string **sessionKey**
- boolean **createDefaultSystemGroup** - True if we should automatically create system groups, false otherwise.

## Returns

- int - 1 on success, exception thrown otherwise.

## 43.25. METHOD: SETDETAILS

Updates the details of a user.

## Parameters

- string **sessionKey**
- string **login** - User's login name.
- struct **user details**
  - string **first\_names** - deprecated, use `first_name`
  - string **first\_name**
  - string **last\_name**
  - string **email**
  - string **prefix**
  - string **password**

## Returns

- int - 1 on success, exception thrown otherwise.



## 43.26. METHOD: SETERRATANOTIFICATIONS

Enables/disables errata mail notifications for a specific user.

### Parameters

- string `sessionKey`
- string `login` - User's login name.
- boolean `value` - True for enabling errata notifications, False for disabling

### Returns

- int - 1 on success, exception thrown otherwise.

## 43.27. METHOD: SETREADONLY

Sets whether the target user should have only read-only API access or standard full scale access.

### Parameters

- string `sessionKey`
- string `login` - User's login name.
- boolean `readOnly` - Sets whether the target user should have only read-only API access or standard full scale access.

### Returns

- int - 1 on success, exception thrown otherwise.

## 43.28. METHOD: USEPAMAUTHENTICATION

Toggles whether or not a user uses PAM authentication or basic RHN authentication.

### Parameters

- string `sessionKey`
- string `login` - User's login name.
- int `pam_value`
  - 1 to enable PAM authentication
  - 0 to disable.

### Returns

- int - 1 on success, exception thrown otherwise.

## APPENDIX A. REVISION HISTORY

<b>Revision 2.0-11</b> Mass publication of all Satellite 5.7 books	<b>Thu Aug 20 2015</b>	<b>Dan Macpherson</b>
<b>Revision 2.0-10</b> Adding Python example	<b>Tue Aug 11 2015</b>	<b>Dan Macpherson</b>
<b>Revision 2.0-9</b> Minor Revisions	<b>Wed May 27 2015</b>	<b>Dan Macpherson</b>
<b>Revision 2.0-8</b> Additional fix (BZ#1203674)	<b>Tue Apr 7 2015</b>	<b>Dan Macpherson</b>
<b>Revision 2.0-7</b> Fixing descriptions for merge actions (BZ#1203674)	<b>Mon Apr 6 2015</b>	<b>Dan Macpherson</b>
<b>Revision 2.0-6</b> Restructuring TOC (BZ#1180456)	<b>Tue Feb 17 2015</b>	<b>Dan Macpherson</b>
<b>Revision 2.0-5</b> Pushing maintenance update for Satellite 5.7	<b>Tue Feb 3 2015</b>	<b>Dan Macpherson</b>
<b>Revision 2.0-4</b> Packaging snapshot versions	<b>Wed Jan 7 2015</b>	<b>Dan Macpherson</b>
<b>Revision 2.0-3</b> Release Candidate for Satellite 5.7	<b>Thu Jan 1 2015</b>	<b>Dan Macpherson</b>
<b>Revision 2.0-2</b> Preparing books for technical review	<b>Mon Dec 8 2014</b>	<b>Dan Macpherson</b>
<b>Revision 2.0-1</b> Remove API reference section; focus on concepts and examples. Add section on read-only API	<b>Fri Nov 7 2014</b>	<b>David O'Brien</b>
<b>Revision 1.0-15</b> Final version of documentation suite	<b>Fri Sep 27 2013</b>	<b>Dan Macpherson</b>
<b>Revision 1.0-14</b> Final changes for Satellite API	<b>Mon Sep 23 2013</b>	<b>Dan Macpherson</b>
<b>Revision 1.0-13</b> Revised Subtitle, Abstract and Preface for all Guides	<b>Tue Sep 10 2013</b>	<b>Dan Macpherson</b>
<b>Revision 1.0-12</b> Minor change on configchannel.getEncodedFileRevision() for BZ#998951	<b>Wed Sep 4 2013</b>	<b>Dan Macpherson</b>
<b>Revision 1.0-11</b> Fixing packages.getPackage() method for BZ#1003565	<b>Wed Sep 4 2013</b>	<b>Dan Macpherson</b>
<b>Revision 1.0-10</b> Elaboration on configchannel.getEncodedFileRevision() for BZ#998951	<b>Wed Sep 4 2013</b>	<b>Dan Macpherson</b>
<b>Revision 1.0-9</b> First implementation of QE Review feedback	<b>Thu Aug 29 2013</b>	<b>Dan Macpherson</b>

<b>Revision 1.0-8</b> Second implementation of tech review feedback	<b>Sun Jul 28 2013</b>	<b>Dan Macpherson</b>
<b>Revision 1.0-7</b> Corrections for BZ#987245	<b>Wed Jul 24 2013</b>	<b>Dan Macpherson</b>
<b>Revision 1.0-6</b> First implementation of tech review feedback	<b>Tue Jul 23 2013</b>	<b>Dan Macpherson</b>
<b>Revision 1.0-5</b> Typo correction	<b>Wed Jul 3 2013</b>	<b>Dan Macpherson</b>
<b>Revision 1.0-4</b> Final beta updates	<b>Wed Jul 3 2013</b>	<b>Dan Macpherson</b>
<b>Revision 1.0-3</b> Beta documentation creation	<b>Wed Jul 3 2013</b>	<b>Dan Macpherson</b>
<b>Revision 1.0-2</b> Minor documentation cleaning	<b>Wed Jul 3 2013</b>	<b>Dan Macpherson</b>
<b>Revision 1.0-1</b> Minor fix to examples	<b>Wed Jul 3 2013</b>	<b>Dan Macpherson</b>
<b>Revision 1.0-0</b> Initial creation and generation of the API Guide	<b>Wed Jul 3 2013</b>	<b>Dan Macpherson</b>