

# Software AG Command Central Help

Version 9.8

April 2015

---

This document applies to Software AG Command Central Version 9.8 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2013-2015 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

# Table of Contents

<b>About this Guide.....</b>	<b>17</b>
<b>Document Conventions.....</b>	<b>17</b>
<b>Online Information.....</b>	<b>18</b>
<b>Understanding Command Central.....</b>	<b>19</b>
<b>Command Central Overview.....</b>	<b>21</b>
About Command Central.....	22
Command Central Features.....	22
Command Central Architecture.....	23
Command Central User Interfaces.....	23
Command Central Server.....	24
Platform Manager.....	24
Platform Manager Installation and Configuration.....	25
Getting Started.....	25
Command Central Terminology.....	26
<b>Using the Command Central Web User Interface.....</b>	<b>29</b>
Accessing Command Central.....	30
Manually Starting and Stopping Command Central and Platform Manager on Windows.....	30
Manually Starting and Stopping Command Central and Platform Manager on Unix.....	30
Troubleshooting Command Central and Platform Manager.....	31
Understanding the Web User Interface.....	31
View Environments.....	31
View Instances.....	32
Monitor Instances.....	33
View Dashboard Information.....	33
View Details of Instances.....	34
View Configuration Parameters.....	35
View Installations.....	35
View Details of an Installation.....	37
View Products in an Installation.....	37
View Fixes Applied to the Products In an Installation.....	37
Manage Lifecycle Actions.....	37
Understanding Icons.....	38
Installation Status Icons.....	38
Instance Status Icons.....	38
Alerts Icons.....	39
<b>Using the Command Line Interface.....</b>	<b>41</b>

Performing Post Installation Configuration.....	42
Customize Configuration Settings.....	42
Define Environment Variables.....	42
Displaying Help for the Command Line Interface.....	43
Executing Command Central Commands.....	44
Executing Platform Manager Commands.....	44
Getting Familiar with Using Commands.....	45
Summary of Commands.....	47
Return Codes from Command Execution.....	58
Options for the Commands.....	59
Common Options.....	59
--accept   -a.....	59
--check-every   -c.....	60
--configuration-file.....	61
--debug   -d.....	62
--error   r.....	63
--expected-values   -e.....	64
--force.....	66
--format   -f.....	67
Arguments for Tab-Separated Values (tsv) and Comma-Separated Values (csv).....	68
--input   -i.....	70
--input-format   -m.....	72
--log   -l.....	73
--media-type   -m.....	74
--output   -o.....	75
--output-format   -f.....	76
--password   -p.....	76
--quiet   -q.....	77
--server   -s.....	77
--ssl-truststore-file.....	78
--ssl-trust-all-hosts.....	79
--ssl-truststore-password.....	79
--username   -u.....	80
--wait   -w.....	80
Invoking Commands from Scripts.....	81
Creating Shell Scripts that Execute Commands.....	81
Creating Ant Scripts that Execute Commands.....	82
Parameters to Use with the ccsetup Task.....	82
Parameters to Use with the cc Task.....	84
<b>Working with Command Central and Platform Manager.....</b>	<b>95</b>
<b>Defining a Managed Landscape.....</b>	<b>97</b>
About Defining a Managed Landscape.....	98
Adding Environments.....	98
Filtering Environments.....	99

---

Editing Environments.....	99
Deleting Environments.....	100
Hiding/Showing Environments Pane.....	100
Administering Installations.....	100
About Administering Installations.....	100
Adding Installations with the Add Installation Wizard.....	101
Adding an Existing Installation.....	101
Installing Platform Manager Remotely.....	102
Installing Platform Manager Remotely On the Command Central Host Server.....	106
Viewing Installations.....	106
Searching for Installations.....	107
Removing Installations.....	107
Linking Installations to Multiple Environments.....	108
Monitoring Installations.....	108
Provisioning Bootstrap Installers Commands.....	109
cc create provisioning bootstrap installers.....	109
cc list provisioning bootstrap installers.....	112
cc exec provisioning bootstrap nodes.....	113
Provisioning Products and Fixes Commands.....	117
cc exec provisioning fixes install.....	117
cc exec provisioning products install.....	119
cc exec provisioning uninstall.....	120
Landscape Commands.....	122
cc add landscape environments nodes.....	122
cc create landscape environments.....	123
cc delete landscape environments.....	125
cc get landscape environments.....	127
cc list landscape environments.....	128
cc remove landscape environments nodes.....	130
cc update landscape environments.....	131
cc create landscape nodes.....	133
cc delete landscape nodes.....	135
cc exec landscape nodes generateNodeId.....	137
cc get landscape nodes.....	139
cc list landscape nodes.....	140
cc update landscape nodes.....	142
Using Command Central to Manage Landscapes with Lower Versions.....	144
About Managing Landscapes with Lower Versions.....	144
Applying Snapshot Templates.....	144
Configuring Authentication Credentials.....	145
<b>Securing the Command Central Landscape.....</b>	<b>147</b>
About Securing the Command Central Landscape.....	148
Setting Up the Administrator User Password for Command Central.....	148

Using a Unix Shell Script to Change the Administrator Password for Command Central.....	149
Setting Up the Administrator User Password for all Platform Managers.....	149
Using a Unix Shell Script to Change the Administrator Password for Platform Manager.....	150
Setting Outbound Authentication.....	150
Changing Authentication Mode.....	151
Changing the Administrator User Password for Managed Products in Command Central.....	151
Changing the Administrator User Password for Managed Products in the Product's User Interface.....	152
Using Unix Shell Scripts to Change Connection Credentials for Managed Products.....	154
Verifying the Outbound Authentication Settings.....	155
Accessing Administrative Interfaces through Command Central.....	155
Working with Configurations that Include Passwords.....	156
Configuring a Shared Secret Globally or for an Environment.....	157
Specify Environment and Global Shared Secrets Using Commands.....	157
Configuring a Shared Secret for a Template.....	158
Security Credentials Commands.....	158
cc add security credentials.....	158
cc add security credentials sharedsecret.....	162
cc delete security credentials.....	163
cc get security credentials.....	164
Securing Communication with the Command Central Server.....	166
Prerequisites to Configuring a Port for SSL.....	166
Configuring SSL Using Command Options.....	166
Configuring SSL Using Configuration Properties Files.....	166
Creating a Custom Command Central Properties File.....	167
Configuring SSL Using Scripts.....	168
Considerations When Using Configuration Properties.....	168
<b>Managing Users, Groups, and Roles.....</b>	<b>169</b>
Managing Command Central Users, Groups, and Roles.....	170
Managing Internally Defined Users in Command Central.....	170
Using Internally Defined User and Group Information.....	171
internaluserrepo Script.....	171
internaluserrepo Exit Codes.....	173
Adding Users to the Internal User Repository.....	174
Deleting Users from the Internal User Repository.....	174
Using Externally Defined User and Group Information.....	175
How Command Central Authenticates Externally Defined Clients.....	175
Overview of Using LDAP.....	176
LDAP Profile Properties.....	176
Configuring LDAP Profile Properties.....	179
Using JAAS with Command Central.....	180
JAAS Configuration File.....	180

Editing the jaas.config File in Command Central.....	181
Editing the jaas.config File Using the cc update configuration Command.....	181
Configuring the jaas.config File to Use LDAP/AD.....	182
Groups.....	183
Default Groups.....	183
Managing Groups.....	183
Editing the groups.txt File in a Text Editor.....	184
Roles.....	184
Default Roles.....	185
Managing Roles.....	185
Editing the roles.txt File in a Text Editor.....	186
<b>Viewing Product Inventory.....</b>	<b>189</b>
About Inventory Management.....	190
Viewing Products in an Installation.....	190
Viewing Fixes Applied to Products in an Installation.....	190
Comparing Product Versions, Fixes, and Configurations.....	190
About Comparing Products.....	190
Comparing Product Versions.....	191
Comparing Fix Levels.....	191
Comparing Configuration Settings.....	192
Inventory Commands.....	192
cc create inventory components attributes.....	192
cc get inventory components.....	194
cc get inventory components attributes.....	196
cc list inventory components.....	198
Specifying Search Criteria for Inventory Commands.....	201
cc list inventory components attributes.....	203
cc update inventory components.....	205
cc update inventory components attributes.....	206
cc delete inventory components attributes.....	208
cc get inventory fixes compare.....	210
cc list inventory fixes.....	211
cc get inventory products.....	213
cc get inventory products compare.....	215
cc list inventory products.....	216
Resources Commands.....	219
cc list resources icons.....	219
<b>Monitoring Instances.....</b>	<b>221</b>
About Monitoring Instances.....	222
Viewing the Status of an Instance or Its Components.....	222
Modifying the Status of an Instance or Its Components.....	222
About Monitoring KPIs.....	223
Viewing Alerts for an Instance or Its Components.....	223
Clearing Alerts for an Instance or Its Components.....	225

About Monitoring-Related Events.....	225
Monitoring KPIs of Platform Manager Instances.....	226
Monitoring Commands.....	227
cc get monitoring.....	227
cc list monitoring alerts.....	230
<b>Administering Product Lifecycle.....</b>	<b>233</b>
About Administering Product Lifecycle.....	234
Lifecycle Actions.....	234
Starting, Stopping, Pausing, Resuming, and Debugging Instances.....	234
Viewing Product Logs.....	235
Viewing the Contents of a Log.....	235
Downloading a Log.....	236
Downloading Multiple Logs.....	236
Lifecycle Commands.....	236
cc exec lifecycle.....	236
Specifying Search Criteria for Lifecycle Commands.....	239
<b>Configuring Products.....</b>	<b>241</b>
Configuring Instances.....	242
Testing a Configuration.....	243
References to File Locations in Product Configuration Files.....	243
Configuration Commands.....	244
cc get configuration common.....	244
cc get configuration compare.....	245
cc create configuration data.....	247
cc delete configuration data.....	251
cc get configuration data.....	253
cc update configuration data.....	256
cc get configuration instances.....	260
cc list configuration instances.....	263
cc get configuration types.....	265
cc list configuration types.....	267
cc exec configuration validation create.....	269
cc exec configuration validation delete.....	272
cc exec configuration validation update.....	274
Instance Management of Installed Products.....	277
About Instance Management of Installed Products.....	277
Instance Management Commands.....	277
cc create instances.....	277
cc delete instances.....	279
cc list instances.....	281
cc list instances supported products.....	282
cc update instances.....	283
<b>Repository Management.....</b>	<b>285</b>

Creating an Image Repository.....	286
Registering a Master Repository.....	287
Deleting a Repository.....	288
Editing Image Repository Details.....	288
Editing Master Repository Details.....	289
Repositories Commands.....	289
cc add repository fixes.....	289
cc add repository products.....	291
cc add repository.....	293
cc delete repository.....	295
cc delete repository with node_alias.....	296
cc delete repositories.....	297
cc exec repository discover.....	298
cc exec repository discover with node_alias.....	299
cc exec repository register.....	301
cc exec repository fixes export.....	302
cc list repository.....	303
cc list repository with node_alias.....	304
cc list repository discover.....	305
cc list repository fixes content.....	307
cc list repository fixes dependencies.....	308
cc list repository fixes readme.....	309
cc list repository platforms.....	310
cc list repository products content.....	311
cc list repository products dependencies.....	313
cc update repository.....	314
cc update repository details.....	315
<b>Template-based Provisioning.....</b>	<b>317</b>
About Template-based Provisioning.....	318
Provisioning Installations.....	318
Creating a Snapshot Template.....	320
Modifying a Snapshot Template.....	321
Applying a Snapshot Template.....	322
Using Default Snapshot Templates.....	324
Considerations When Using Default Snapshot Templates.....	326
Provisioning Environments.....	327
Composite Template Definition.....	328
Using Default Composite Templates.....	334
Considerations When Using Default Composite Templates.....	337
Creating Custom Composite Templates.....	337
Template Commands.....	338
cc create templates snapshot.....	338
cc delete templates snapshot.....	342
cc delete templates composite.....	343

cc exec templates snapshot apply.....	343
cc exec templates composite apply.....	348
cc exec templates snapshot import.....	349
cc exec templates composite import.....	350
cc export templates snapshot.....	351
cc list templates composite.....	353
cc list templates snapshot.....	353
<b>The Java Service Wrapper.....</b>	<b>355</b>
The Java Service Wrapper Configuration Files.....	356
Specifying the JDK or JRE.....	357
JVM Configuration.....	357
JVM Configuration Properties.....	357
Configuring the JVM in Command Central.....	358
The Wrapper Log File.....	359
Logging Properties.....	359
Fault Monitoring.....	359
Generating a Thread Dump.....	360
<b>Managing Command Central Licenses and Product License Reports.....</b>	<b>361</b>
Command Central License Overview.....	362
Viewing the Command Central License Information.....	362
Changing a Command Central License Key.....	362
Renewing a Command Central License Key.....	363
Managing Product License Reports.....	363
Creating a License Report.....	364
Viewing Details About License Reports.....	364
Sorting License Reports.....	365
Downloading License Reports.....	365
Deleting License Reports.....	365
Managing License Manifests.....	365
Adding License Manifests.....	365
Viewing Details About License Manifests.....	366
Downloading License Manifests.....	366
Deleting License Manifests.....	366
License Reports Commands.....	367
cc create license-tools reports snapshot.....	367
cc delete license-tools reports snapshot.....	367
cc delete license-tools reports snapshot reportid.....	368
cc list license-tools reports snapshot.....	369
cc get license-tools reports snapshot reportid.....	370
cc get license-tools reports snapshot output PDF.....	371
cc get license-tools reports snapshot output XML.....	372
cc create license-tools reports installation.....	372
cc delete license-tools reports installation reportid.....	373
cc delete license-tools reports installation.....	374

cc get license-tools reports installation output PDF.....	375
cc get license-tools reports installation output XML.....	376
cc get license-tools reports installation disclaimer.....	376
cc list license-tools reports installation.....	377
cc list license-tools reports.....	378
cc list license-tools metering products.....	379
cc add license-tools manifests.....	379
cc delete license-tools manifests.....	380
cc get license-tools manifests output xml.....	381
cc list license-tools manifests.....	382
<b>Viewing Jobs.....</b>	<b>383</b>
Diagnostics Logs Commands.....	384
cc get diagnostics logs.....	384
cc get diagnostic logs export file.....	388
cc list diagnostics logs.....	389
Jobmanager Jobs Commands.....	391
cc list jobmanager jobs.....	391
<b>Understanding Product-specific Administration.....</b>	<b>393</b>
<b>Administering webMethods Broker.....</b>	<b>395</b>
About webMethods Broker Administration.....	396
Configuring Broker Server License.....	396
Configuring SSL in Broker Server.....	397
Retrieving Configuration Details of Broker Server Base Port.....	398
Pausing and Resuming Message Publishing in Broker Servers.....	399
Using the Administration Link of Broker Server.....	400
Configuring the Host and Port of My webMethods Server.....	400
Pre-requisites for Viewing the Broker Server Details Page in My webMethods.....	400
Viewing the Broker Server Details Page in My webMethods.....	401
Monitoring webMethods Broker KPIs.....	401
Overview.....	401
Storage Utilization KPI.....	402
Marginal, Critical, and Maximum Values for Broker Server's Storage Utilization.....	402
Storage Utilization Display.....	402
Memory Utilization KPI.....	403
Marginal, Critical, and Maximum Values for Memory Utilization.....	404
Stalled Queues KPI.....	404
webMethods Broker and the Command Line Interface.....	405
Commands that webMethods Broker Supports.....	405
Configuration Types that webMethods Broker Supports.....	407
Lifecycle Actions for Broker Server.....	408
Monitoring Run-time Statuses for webMethods Broker.....	409
Monitoring Run-time States for webMethods Broker.....	410

---

<b>Administering CentraSite</b> .....	<b>413</b>
About Administering CentraSite.....	414
Viewing CentraSite Components.....	414
Viewing CentraSite Registry Repository (CRR) and CentraSite Application Server Tier (CAST).....	414
CentraSite and the Command Line Interface.....	415
Commands that CentraSite Registry Repository Supports.....	415
Commands that CentraSite Application Server Tier Supports.....	416
Lifecycle Actions for CentraSite Registry Repository.....	416
Run-time Monitoring Statuses for CentraSite Registry Repository.....	417
<b>Cloud Factory Services Overview</b> .....	<b>419</b>
Configuring Cloud Factory Services Accounts.....	420
Adding an Amazon Elastic Compute Cloud Account.....	420
Adding a Rackspace Account.....	421
Adding a VMware vSphere Account.....	422
Editing Accounts.....	423
Deleting Accounts.....	424
Cloud Factory Services Commands.....	424
cc create cfs instances.....	424
cc delete cfs instances.....	427
cc exec cfs instances.....	428
cc get cfs instances.....	430
cc list cfs accounts.....	431
cc list cfs images.....	432
cc list cfs instances.....	434
cc list cfs vendors.....	435
<b>Configuring Integration Server</b> .....	<b>437</b>
About Integration Server Configuration Types.....	438
Working with Integration Server Configuration Types.....	438
Configuring Integration Server Ports.....	439
About Ports.....	439
Configuring Ports.....	440
Testing Ports.....	445
Setting the Primary Port.....	445
Editing Port Information.....	445
Enabling and Disabling Ports.....	446
Monitoring KPIs of Integration Server Instances.....	446
Integration Server and the Command Line Interface.....	447
Command Central Command Line Tool Upgrade.....	447
Commands that Integration Server Supports.....	447
Configuration Types that IntegrationServer-instanceName Supports.....	450
Lifecycle Actions for Integration Server.....	453
Integration Server Instance Management.....	454

Run-time Monitoring Statuses for IntegrationServer-instanceName.....	456
Run-time Monitoring States for IntegrationServer-instanceName.....	457
<b>Configuring My webMethods Server.....</b>	<b>459</b>
Configuring My webMethods Server Ports.....	460
Editing Port Settings.....	460
Configuring My webMethods Server Email.....	461
Working with My webMethods Server Environment Variables.....	462
Considerations when Configuring My webMethods Server Variables.....	462
Configuring My webMethods Server Variables.....	462
Monitoring KPIs of My webMethods Server Instances.....	463
Using Trusted Authentication to Connect to My webMethods Server.....	464
My webMethods Server and the Command Line Interface.....	464
Commands that My webMethods Server Supports.....	464
Configuration Types that My webMethods Server-ENGINE Supports.....	467
Lifecycle Actions for My webMethods Server-ENGINE.....	470
My webMethods Server Instance Management.....	470
Creating a My webMethods Server Instance.....	470
Updating a My webMethods Server Instance.....	474
Deleting a My webMethods Server Instance.....	475
Run-time Monitoring Statuses for My webMethods Server-ENGINE.....	475
Run-time Monitoring States for My webMethods Server.....	476
<b>Configuring OSGi Profiles.....</b>	<b>479</b>
Protocols that Command Central Supports in OSGi Profiles.....	480
Default Ports.....	481
Products that Support Port Configuration in OSGi Profiles.....	481
Port Authentication.....	482
Configuring Ports in OSGi Profiles.....	483
Adding Ports.....	483
Viewing the Port Settings.....	487
Editing and Testing OSGi Port Information.....	488
Deleting a Port.....	488
OSGi Components and the Command Line Interface.....	488
Configuration Types that the OSGi Profile Components Supports.....	488
Lifecycle Actions for the OSGi Profile Components.....	495
Run-time Monitoring Statuses for the OSGi Profile Components.....	496
Run-time Monitoring States for OSGi Profile Components.....	497
Configuring the Command Central OSGi Profile Component.....	498
Configuration Types that OSGi-CCE-ENGINE Supports.....	498
Lifecycle Actions for OSGi-CCE-ENGINE.....	499
Run-time Monitoring Statuses for OSGi-CCE-ENGINE.....	499
Configuring the Platform Manager OSGi Profile Component.....	500
Configuration Types that OSGi-SPM-ENGINE Supports.....	500
Run-time Monitoring Statuses for OSGi-SPM-ENGINE.....	501
Run-time Monitoring States for OSGi-SPM-ENGINE.....	501

Configuring the Tomcat OSGI Profile Components.....	503
Lifecycle Actions for OSGI-*-TOMCAT-ENGINE.....	503
Run-time Monitoring Statuses for OSGI-*-TOMCAT-ENGINE.....	504
<b>Administering Task Engine.....</b>	<b>505</b>
Overview.....	506
Commands that Task Engine Supports.....	506
Configuration Types that Task Engine Supports.....	508
Run-Time Monitoring Statuses for Task Engine.....	508
<b>Administering Universal Messaging.....</b>	<b>511</b>
About Administering Universal Messaging.....	512
Universal Messaging Configuration Types.....	512
Universal Messaging License Configuration.....	513
Universal Messaging Memory Configuration.....	513
Universal Messaging Ports Configuration.....	513
Port Configuration Attributes.....	513
Basic Port Connection Attributes.....	513
Port Security Attributes.....	514
Universal Messaging Authentication Configuration.....	515
Universal Messaging Users Configuration.....	516
Universal Messaging Cluster Management.....	516
Before You Create a Universal Messaging Cluster.....	516
Cluster Configuration Tasks Supported.....	516
Cluster Configuration Fields.....	517
Universal Messaging Inventory.....	517
Universal Messaging Lifecycle Actions.....	517
Universal Messaging Run-time Monitoring Statuses.....	518
Universal Messaging KPIs.....	518
Universal Messaging and the Command Line Interface.....	519
Commands that Universal Messaging Server Instance Supports.....	519
Configuration Types that Universal Messaging Server Instance Supports.....	523
Universal Messaging Server Instance Inventory.....	524
Universal Messaging Server Instance License Management.....	525
Universal Messaging Server Instance Memory Configuration.....	525
Universal Messaging Server Instance Management.....	526
Universal Messaging Server Instance User Management.....	527
internaluserrepo Script.....	529
Universal Messaging Cluster Management.....	530
Before You Create or Update a Universal Messaging Server Instance Cluster.....	530
Create a Universal Messaging Cluster.....	530
Get Universal Messaging Cluster Details.....	532
Update Universal Messaging Cluster.....	532
Delete Universal Messaging Cluster.....	533
Universal Messaging Server Instance Lifecycle Actions.....	533
Run-time Monitoring Statuses for a Universal Messaging Server Instance.....	534

---

Run-time Monitoring States for a Universal Messaging Server Instance.....	535
<b>Administering WmTaskClient.....</b>	<b>537</b>
Overview.....	538
Commands that WmTaskClient Supports.....	538
Configuration Types that WmTaskClient Supports.....	540
Run-Time Monitoring Statuses for WmTaskClient.....	541
<b>Administering Terracotta.....</b>	<b>543</b>
Commands that Terracotta Supports.....	544
Configuration Types that Terracotta Supports.....	545
Lifecycle Actions for Terracotta.....	545
Run-time Monitoring Statuses for Terracotta.....	546
Run-time Monitoring States for Terracotta.....	546
<b>Command Central Task Quick Reference.....</b>	<b>549</b>
<b>Working with Authentication between Command Central and Managed Products.....</b>	<b>550</b>
<b>Working with Environments.....</b>	<b>550</b>
<b>Working with Installations.....</b>	<b>552</b>
<b>Working with Templates.....</b>	<b>555</b>
<b>Working with Instances.....</b>	<b>555</b>
<b>Working with Logs.....</b>	<b>559</b>
<b>Working with Product Comparisons.....</b>	<b>560</b>
<b>Working with Product Inventory.....</b>	<b>561</b>
<b>Working with Repositories.....</b>	<b>562</b>
<b>Working with Configuration Types.....</b>	<b>564</b>
<b>Working with Ports.....</b>	<b>565</b>
<b>Working with KPIs.....</b>	<b>566</b>
<b>Working with Security Credentials.....</b>	<b>566</b>
<b>Working with Licenses.....</b>	<b>567</b>
<b>Supported Functions and Configurations for Different Product Versions.....</b>	<b>571</b>
<b>Broker.....</b>	<b>572</b>
<b>Business Rules.....</b>	<b>573</b>

---

<b>CentraSite</b> .....	<b>574</b>
<b>Cloud Factory</b> .....	<b>574</b>
<b>Event-Driven Architecture/NERV</b> .....	<b>575</b>
<b>Infrastructure Data Collector</b> .....	<b>576</b>
<b>Integration Server</b> .....	<b>577</b>
<b>Monitor</b> .....	<b>579</b>
<b>My webMethods Server</b> .....	<b>580</b>
<b>OneData</b> .....	<b>582</b>
<b>Optimize User Interface</b> .....	<b>583</b>
<b>Platform Manager</b> .....	<b>584</b>
<b>Task Engine</b> .....	<b>585</b>
<b>Terracotta</b> .....	<b>586</b>
<b>Universal Messaging</b> .....	<b>587</b>

---

## About this Guide

---

This guide provides information about working with Command Central and Platform Manager to configure, manage, and administer one or more installations of the webMethods product suite in your enterprise.

## Document Conventions

---

Convention	Description
<b>Bold</b>	Identifies elements on a screen.
Narrowfont	Identifies storage locations for services on webMethods Integration Server, using the convention <i>folder.subfolder:service</i> .
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
<i>Italic</i>	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
Monospace font	Identifies text you must type or messages displayed by the system.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the   symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

---

## Online Information

---

### Software AG Documentation Website

You can find documentation on the Software AG Documentation website at <http://documentation.softwareag.com>. The site requires Empower credentials. If you do not have Empower credentials, you must use the TECHcommunity website.

### Software AG Empower Product Support Website

You can find product information on the Software AG Empower Product Support website at <https://empower.softwareag.com>.

To submit feature/enhancement requests, get information about product availability, and download products, go to [Products](#).

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the [Knowledge Center](#).

### Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at <http://techcommunity.softwareag.com>. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

# I Understanding Command Central

---

■ Command Central Overview .....	21
■ Using the Command Central Web User Interface .....	29
■ Using the Command Line Interface .....	41



# 1 Command Central Overview

---

■ About Command Central .....	22
■ Command Central Features .....	22
■ Command Central Architecture .....	23
■ Getting Started .....	25
■ Command Central Terminology .....	26

## About Command Central

---

Software AG Command Central is a tool that release managers, infrastructure engineers, system administrators, and operators can use to perform administrative tasks from a single location. Command Central can assist with the following configuration, management, and monitoring tasks:

- Infrastructure engineers can see at a glance which products and fixes are installed, where they are installed, and compare installations to find discrepancies.
- System administrators can configure environments by using a single web user interface or command-line tool. Maintenance involves minimum effort and risk.
- Release managers can prepare and deploy changes to multiple servers using command-line scripting for simpler, safer lifecycle management.
- Operators can monitor server status and health, as well as start and stop servers from a single location. They can also configure alerts to be sent to them in case of unplanned outages.

## Command Central Features

---

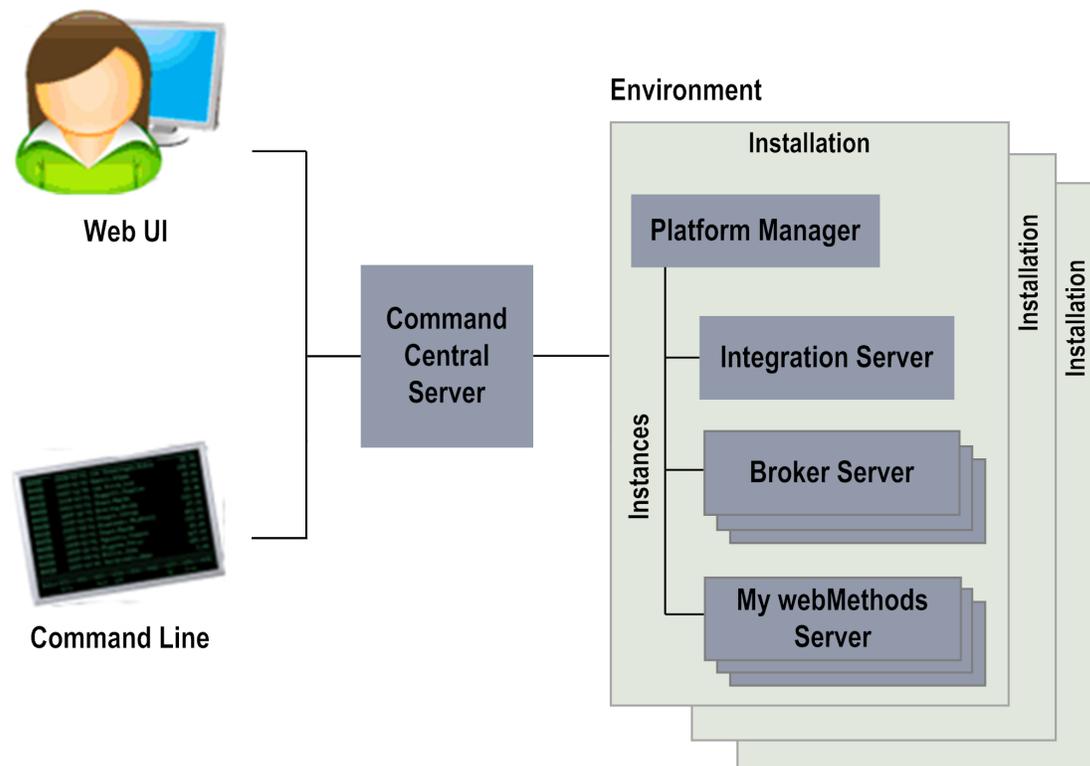
Using Command Central, you can administer hundreds of managed product installations in your IT landscape from a central location. Command Central supports the following operations:

- Provision new and update existing environments using default or customized composite templates that you import and apply using the Command Central command line tool.
- Create a template from an existing managed installation and apply the template to another managed installation to repeat the same set of products, fixes and configuration parameters.
- Create and manage multiple instances of a product in the same installation.
- Secure communication to the Command Central server.
- View inventory of webMethods product installations, versions, and fixes.
- Compare the versions of the products installed in different installations.
- Compare the fixes applied to products in different installations.
- Configure configuration settings of product instances, such as ports, licenses, and alerts.
- Compare the configuration settings of product instances running on different installations.

- Perform lifecycle operations such as start, stop, restart, pause, resume, and debug on runtime processes.
- Monitor the health of product installations.
- Monitor run-time status, KPIs, and alerts of product instances.

## Command Central Architecture

Command Central is built on top of Software AG Common Platform, which uses the OSGi (Open Services Gateway Initiative) framework. Product-specific features are in the form of plug-ins.



## Command Central User Interfaces

Command Central users can communicate with Command Central Server using one of the following interfaces:

- Graphical web user interface, for administering products using the web
- Command line interface, for automating administrative operations

## Command Central Server

Command Central Server accepts administrative commands that users submit through one of the three user interfaces and then directs the commands to the respective Platform Manager for execution.

An *installation* in Command Central is a set of Software AG products and fixes installed in the same installation directory. Products that Command Central manages are referred to as managed products throughout this help.

Command Central can manage one or more installations of the following products and product components:

- Platform Manager
- Command Central
- webMethods Broker
- webMethods Integration Server
- My webMethods Server
- CentraSite
- Universal Messaging
- Terracotta
- Monitor
- Business Rules
- My webMethods user interface for Optimize
- Task Engine
- Event Routing

Command Central provides a common location for configuring managed products installed in different environments.

## Platform Manager

Software AG Platform Manager manages Software AG products. Platform Manager enables Command Central to centrally administer the lifecycle of managed products. In a host machine, you might have multiple Software AG product installations. For each Software AG product installation, you need a separate Platform Manager to manage the installed products. For example, if you have these installation directories in a host machine:



- Configure the product instances
- View the inventory of products and fixes
- Perform lifecycle operations such as start, stop, restart, pause, resume, and debug on run-time processes
- Monitor the product instances
- Compare configuration settings of the products installed
- Compare the versions of the products installed
- Compare the fixes applied to the products

## Command Central Terminology

---

The following table defines common Command Central terms.

Term	Description
Component	<p>An independent module that runs within a process but has its own configurable elements. A component can be started and stopped, administered, and monitored separately. The lifecycle of a component is dependent on the parent component. That is, a component stops if its parent component stops.</p> <p>For example, Platform Manager Core Services is a component of Platform Manager.</p>
Composite Template	<p>A declarative definition of an environment that describes how to set up the installations, layers, templates, repositories, and configuration required for creating a new environment or updating an existing one.</p>
Environment	<p>A collection of layers and the installations that the layers are hosted on, logically grouped together for easier management.</p> <p>For example, you can group the layers used for testing under an environment called Testing.</p>
Installation	<p>A set of Software AG products and fixes installed in the same installation directory on a local or remote machine and managed under a single Platform Manager instance. An installation either already exists</p>

---

<b>Term</b>	<b>Description</b>
	<p>on the host machine, or is installed automatically (bootstrapped) by Platform Manager.</p> <p>An installation can host different layers.</p>
Instance	<p>A single copy of a running product. An instance of a product is defined by its configuration settings. Some products might have multiple instances in one installation.</p> <p>For example, Broker Server and My webMethods Server are instances.</p>
Landscape	<p>A collection of all environments that are managed under a single Command Central instance.</p> <p>Often the landscape design decisions are based on security requirements and physical network topology.</p>
Layer	<p>A collection of instances that have the same logical role. For example, an Integration Server cluster is a layer.</p> <p>Instances grouped logically under the same layer are configured in an identical way.</p> <p>A layer belongs to a single environment and is hosted by one or more installations.</p>
Platform Manager	<p>An architectural component of Command Central that provides a common remote management interface to a Software AG product installation.</p>
Snapshot Template	<p>A definition of products, fixes, instances, configuration, and files that are installed or deployed on a single installation. Snapshot templates are either default templates provided in Command Central, or templates created from existing managed installations using Command Central.</p>
Software AG Common Platform	<p>A Java run-time environment based on the OSGi framework. It provides a standard platform on which to run Software AG products and the enterprise applications that you develop around those products.</p> <p>Software AG Common Platform hosts Software AG products such as Integration Server, My webMethods Server, and Command Central.</p>

<b>Term</b>	<b>Description</b>
	Common Platform does not host native applications such as Broker Server and Universal Messaging.

## 2 Using the Command Central Web User Interface

---

■ Accessing Command Central .....	30
■ Understanding the Web User Interface .....	31
■ Understanding Icons .....	38

## Accessing Command Central

---

After you install Software AG Command Central using Software AG Installer, Platform Manager starts automatically as a service in the port you configured during installation. You start Command Central by starting the Windows service for Command Central. You can access the Command Central web user interface from all supported web browsers.

For information about how to install and configure Command Central, see *Installing webMethods and Intelligent Business Operations Products*.

To access the Command Central web user interface, specify the following URL in your browser:

```
http://hostname :port /cce/web
```

For the *hostname*, specify `localhost` or the name of the host machine where you have installed Command Central.

For the *port*, specify the port number where the Command Central instance is running.

## Manually Starting and Stopping Command Central and Platform Manager on Windows

You can start Command Central or Platform Manager by starting the Windows service for Command Central or Platform Manager.

To shut down the Command Central or Platform Manager servers, stop the Windows service for Command Central or Platform Manager.

## Manually Starting and Stopping Command Central and Platform Manager on Unix

To start Command Central or Platform Manager, execute the startup `.sh` script located in the following directories:

- For Command Central: *Software AG\_directory/profiles/CCE/bin*
- For Platform Manager: *Software AG\_directory/profiles/SPM/bin*

To shut down the Command Central or Platform Manager servers, execute the shutdown `.sh` script located in the following directories:

- For Command Central: *Software AG\_directory/profiles/CCE/bin*
- For Platform Manager: *Software AG\_directory/profiles/SPM/bin*

## Troubleshooting Command Central and Platform Manager

Use the console.bat | sh script to troubleshoot Command Central or Platform Manager. The console.bat | sh script is located in the following directories:

- For Command Central: *Software AG\_directory*\profiles\CCE\bin
- For Platform Manager: *Software AG\_directory*\profiles\SPM\bin

The server logs are located in the following directories:

- For Command Central: *Software AG\_directory*\profiles\CCE\logs
- For Platform Manager: *Software AG\_directory*\profiles\SPM\logs

## Understanding the Web User Interface

This section describes the Command Central web user interface. Software AG recommends you use either Chrome or Firefox browser.

When you use Internet Explorer 9, in the **General** tab of Internet Options, select the **Every time I visit the webpage** option under **Check for newer versions of stored pages** in the Browsing history settings. Otherwise, Internet Explorer 9 browser might not display the newly added environment or installation when you refresh the browser after you add an environment or installation.

**Note:** To terminate the session, close your web browser completely. Closing only the browser tab will not end the session.

## View Environments

Use the Environments pane in the Command Central home page to view the environments that you have defined for administration through Command Central.

In the Environments pane, you can view, add, delete, modify, and search for environments.

The following table describes the fields and icons displayed in the Environments pane.

Icon/Field	Description
	Type the filter criteria for searching the environments in the <b>Search Environments</b> field. The <b>Environments</b> pane lists only those environments with names that match the search criteria.

Icon/Field	Description
	Click  <b>Clear Filter</b> to clear the search filter and display all the environments.
<b>All</b>	Click <b>All</b> to view all the installations defined in Command Central. This is the default environment. This environment contains all the installations that can be administered by Command Central. The <b>All</b> environment includes all the installations grouped under other environments, as well as the installations that are not grouped under any environment.
	Click  <b>Add Environment</b> to add a new environment for administration through Command Central.
	Click  <b>Remove Environment</b> to remove the selected environment from Command Central administration.
	Click  <b>Edit Environment</b> to edit the details of the selected environment.

## View Instances

Use the Instances tab to view the details of the instances in the selected environment.

In the Instances tab, you can view instances and components, configure instances, view and change the status of instances, view alerts, and search for instances.

The following table describes the fields and icons displayed in the Instances tab.

Icon/Field	Description
	Type the filter criteria for searching the instances in this field. The Instances tab lists only the instances with names that match the search criteria.
	Click  <b>Clear Filter</b> to clear the search filter and list all the instances belonging to the selected installation.
	Click  <b>Compare Configuration</b> to select and compare the configurations of multiple instances.

Icon/Field	Description
	Click  to query the Platform Manager and refresh the changes made to the instances.
<b>Name [Count]</b>	Display name of the instance and the total number of child instances.  Expand the instance node to view the list of child instances.
<b>Component</b>	Component name or the product code.
<b>Status</b>	Indicates whether the installation is Online, Failed, Starting, Stopped, Stopping, Unknown, or Unresponsive. For more information about the instance status, see <a href="#">Viewing the Status of an Instance or Its Components</a> .  Click and select a lifecycle action to change the status.
<b>Alerts</b>	Displays an alert flag if there is any alert for the component.
<b>Installation</b>	Name of the installation where the instance is installed.
<b>Host</b>	Name of the host used by the instance.

## Monitor Instances

Use the Overview tab of an instance to view the details about the instance such as the status, alerts, host name, and the installation alias. In addition, you can monitor the status and KPIs of the instance.

Click the name of an instance in the **Instances** tab, and then click the **Overview** tab to view the details about the instance and monitor the instance.

Click the administration link in the **Overview** tab and use the individual product interface to administer the products.

## View Dashboard Information

Use the Dashboard panel in the Overview tab of an instance to view the following information.

Field	Description
<b>Status</b>	Current status of the instance or component. For more information, see <a href="#">Viewing the Status of an Instance or Its Components</a> .
<b>Alerts</b>	Number of alerts. For more information, see <a href="#">Viewing Alerts for an Instance or Its Components</a> .
<b>Monitoring</b>	Key performance indicators. For more information, see <a href="#">About Monitoring KPIs</a> .

## View Details of Instances

Use the Details panel in the Overview tab of an instance to view the following information.

Field	Description
<b>Display Name</b>	<p>Display name of the instance or component. You can edit the display name in line.</p> <p>Click  to modify the icon defined for the instance or component.</p>
<b>Component</b>	Name of the component.
<b>Host Name</b>	Name of the host machine where the instance or component is installed.
<b>Authentication</b>	<p>Authentication mode for administering the instance. The default is Fixed user.</p> <p>Click  to edit the user name and password for fixed user authentication.</p>
<b>Installation Name</b>	Name of the installation where the instance or component is installed.
<b>Installation Alias</b>	Alias name of the installation where the instance or component is installed.
<b>Attributes</b>	List of the user-defined search attributes for the instance component.

Field	Description
	<p>You use the search attributes to search or filter run-time components. You can also use the search attributes to identify the run-time components for which to execute an operation. For more information about the Command Line tool search attributes commands, see <i>Software AG Command Central and Software AG Platform Manager Command Reference</i>.</p> <p>To add a search attribute, click  and specify a name and matching value for the search attribute. Click anywhere outside of the Attributes table or press <b>Enter</b>.</p> <p><b>Note:</b> To add a new search attribute successfully you must specify values in both <b>Name</b> and <b>Value</b> columns.</p> <p>To edit data for a search attribute, click the field for the name or value of the search attribute that you want to change and specify a new value. Click anywhere outside of the Attributes table or press <b>Enter</b>.</p> <p>To delete a search attribute, select the name or value of the search attribute and click .</p>

## View Configuration Parameters

Use the Configuration tab of an instance to configure its parameters such as ports, licenses, and emails. You can also configure the OSGi profiles of components.

Click the name of an instance in the **Instances** tab, and then click the **Configuration** tab, to configure the parameters of the instance.

## View Installations

Use the Installations tab to view the details of all the installations that are part of the selected environment.

Command Central polls Platform Manager every 30 seconds to get the current status and alerts for the products in the installations when you view the Installations tab.

Icon/Field	Description
	Type the filter criteria for searching the installations in this field. The Installations tab lists only the installations with names that match the search criteria.
	Click  <b>Clear Filter</b> to clear the search filter and list all the installations of the selected environment.
	Click  <b>Add Installation</b> to define an installation that you want to administer through Command Central. This installation will be grouped under the selected environment. If you have not specified any environment, the installation is added to the default <b>All</b> environment.
	Click  <b>Remove Installation</b> to remove the selected installation from the specified environment. The removed installation will be listed under <b>All</b> environment. If you do not want an installation to be administered by Command Central, you must explicitly remove it from the <b>All</b> environment.
	Click  to select these options: <ul style="list-style-type: none"> <li>■ <b>Compare Products</b></li> <li>■ <b>Compare Fixes</b></li> </ul>
	Click  to query the Platform Manager and refresh the changes made to the installations.
<b>Name [Count]</b>	Display name and the total number of products installed in the installation node.  Expand the installation node to view the list of products installed in the installation node.  Click an installation node to view the Overview, Products, and Fixes tabs.
<b>Status</b>	Indicates whether the installation is online or offline. Offline status indicates that the Platform Manager is not responding.

---

Icon/Field	Description
Host	Name of the host used by the installation node.
Port	Port number used by the installation node.
Code	Product code of the product in the installation node. Expand the installation node to view the list of products installed in the installation node.
Version	Version number of the Platform Manager or the product.

## View Details of an Installation

Click the name of an installation in the **Installations** tab to view the **Overview** tab of the installation.

Use the Overview tab of an installation to view installation details, monitor the operating system KPIs such as utilization of the storage and memory, administer instances, and compare the configurations of instances belonging to that installation.

Command Central polls Platform Manager every 30 seconds to get the current status of the instances in the installation when you are viewing the Overview tab without navigating away from the tab.

## View Products in an Installation

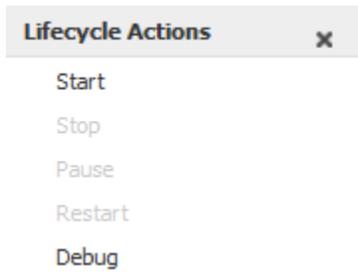
In the **Installations** tab, click the name of an installation and then click the **Products** tab, to view details about the products installed in the selected installation, including the product's name, code, version, mechanism used to install the product, and date and time the product was installed.

## View Fixes Applied to the Products In an Installation

In the **Installations** tab, click the name of an installation and then click the **Fixes** tab, to view the details of the fixes applied to the managed products in the selected installation.

## Manage Lifecycle Actions

Use the Lifecycle Actions dialog to administer the managed products. To view the Lifecycle Actions dialog, click the status of an instance listed in the **Instances** tab or in the **Overview** tab of the instance. For information about the lifecycle actions, see [Administering Product Lifecycle](#).



## Understanding Icons

This section describes the icons used to identify the status of installations, instances, and alerts.

### Installation Status Icons

The installation status indicates whether Command Central is able to connect to the installation using the Platform Manager of that installation.

Icon	Status	Indicates
	Online	Command Central is able to connect to the installation host and port.
	Offline	Command Central cannot connect to the installation host and port.

### Instance Status Icons

The instance status indicates whether the instance is currently running, started, stopped, or unresponsive.

Icon	Status	Indicates...
	Online	The instance or instance component is currently running and the ping operation succeeds.
	Failed	The instance or instance component is not running and the ping operation fails.

Icon	Status	Indicates...
	Starting	The instance or instance component is starting.
	Stopped	The instance or instance component has stopped.
	Stopping	The instance or instance component is stopping.
	Paused	The instance or instance component has paused.
	Unknown	The status of the instance or instance component cannot be determined.
	Unresponsive	The ping operation fails, but other indicators such as the process-id file indicate that the instance or instance component is running.

## Alerts Icons

Alerts are raised or disabled when any of the following condition occurs.

- The status of an instance or instance component changes.
- The value of a KPI (key performance indicator) changes.

Icon	Indicates
	Instance warning or error.
	Instance information.



---

# 3 Using the Command Line Interface

---

■ Performing Post Installation Configuration .....	42
■ Displaying Help for the Command Line Interface .....	43
■ Executing Command Central Commands .....	44
■ Executing Platform Manager Commands .....	44
■ Getting Familiar with Using Commands .....	45
■ Summary of Commands .....	47
■ Return Codes from Command Execution .....	58
■ Options for the Commands .....	59
■ Invoking Commands from Scripts .....	81

**Important:** The command-line interface in Command Central and Platform Manager version 9.8 and above might not be fully compatible with earlier versions. To use version 9.8 and above, you might need to make changes to the scripts that you developed with earlier versions.

## Performing Post Installation Configuration

This topic contains tasks you can perform after installation.

### Customize Configuration Settings

You can customize the default configuration settings for the Command Central command line interface, such as Command Central or Platform Manager endpoint server address, authentication credentials, and SSL certificate configuration. You update the Command Central command line interface configuration settings in the *Software AG\_directory*\CommandCentral\client\conf\cc.properties file.

#### To change the default values for the configuration properties listed in the cc.properties file

1. Go to the *Software AG\_directory* \CommandCentral\client\conf directory.
2. Save a copy of the cc.properties file in your *Software AG\_directory* and set the file permissions of the copied file to deny access to other users.
3. Open the copy of the cc.properties file in a text editor.
4. Edit the configuration properties settings as required.
5. Save the copied file.

You can store copies of the cc.properties file with different configuration settings in different directories and use the `--configuration-file` command option to point to the cc.properties file that you require. For example:

```
cc list inventory components --configuration-file c:\temp\cc.properties
```

### Define Environment Variables

**Important:** You can either configure the Command Central or Platform Manager endpoint server address and authentication credentials in the *Software AG\_directory*\CommandCentral\client\conf\cc.properties file, or set the CC\_SERVER and CC\_USERNAME environment variable. Software AG recommends that you configure the endpoint URLs and user name in the cc.properties file as described in "[Customize Configuration Settings](#)" on page 42.

- Optional. Define environment variables so that you can invoke Command Central and Platform Manager commands from any location on the machine. To do so:

1. Set the `CC_CLI_HOME` environment variable to the following directory:  
*Software AG\_directory*\CommandCentral\client

#### Examples

- Windows: `set CC_CLI_HOME=C:\SoftwareAG\CommandCentral\client`
- UNIX: `export CC_CLI_HOME="/opt/SoftwareAG/CommandCentral/client"`

2. Add "`$CC_CLI_HOME/bin`" to the `PATH` environment variable.

#### Examples

- Windows: `set PATH=%PATH%;%CC_CLI_HOME%\bin`
- UNIX: `export PATH="$PATH:$CC_CLI_HOME/bin"`
- Optional. Set `CC_SERVER` environment variable to either a Command Central or Platform Manager server endpoint address. By doing so, if you omit the `{--server | -s}` option from a command, the command uses the value you specify for the `CC_SERVER` variable. For more information, see "[--server | -s](#)" on page 77.

Note that Software AG recommends that you set the endpoint server address in the `server` property in the `cc.properties` file instead of using the `CC_SERVER` environment variable.

#### Examples

- Windows: `set CC_SERVER=http://rubicon:8090/cce`
- UNIX: `export CC_SERVER="http://rubicon:8090/cce"`
- Optional. You can set `CC_USERNAME` environment variable to a user name. By doing so, if you omit the `{--username | -u}` option from a command, the command uses the value you specify for the `CC_USERNAME` variable. For more information, see "[--server | -s](#)" on page 77.

Note that Software AG recommends that you set the user name in the `username` property in the `cc.properties` file instead of using the `CC_USERNAME` environment variable.

#### Examples

- Windows: `set CC_USERNAME=Administrator`
- UNIX: `export CC_USERNAME="Administrator"`

## Displaying Help for the Command Line Interface

You can display help for the command line interface tool from the command prompt.

---

**To display help for the command line interface tool.**

- To display general help that includes operations and common options, enter `cc` with no other arguments. For example:

```
cc
```

- To display a list of Command Central commands, including the syntax of the commands, use the `{--help | -h}` option. Also include the `{--server | -s}` option to identify a Command Central server. For example:

```
cc --help --server http://rubicon:8090/cc
```

**Note:** If you omit the `{-server | -s}` option, the command uses the value from the `CC_SERVER` environment variable.

- To display a list of Platform Manager commands, including the syntax of the commands, use the `{--help | -h}` option. Also include the `{--server | -s}` option to identify a Platform Manager server. For example:

```
cc --help --server http://spm:8092/spm
```

**Note:** If you omit the `{-server | -s}` option, the command uses the value from the `CC_SERVER` environment variable.

---

## Executing Command Central Commands

---

**To execute a Command Central command:**

1. From the command prompt, change directory to the following location where the executable files for the Command Central commands reside:

```
Software AG_directory\CommandCentral\client\bin
```

**Note:** This step is not necessary if you have set the `CC_CLI_HOME` environment variable and included it on the `PATH` environment variable. For more information, see ["Performing Post Installation Configuration" on page 42](#).

2. Enter the command you want to execute.

For example, to list products that Command Central manages, enter:

```
cc list inventory products
```

---

## Executing Platform Manager Commands

---

There are no separate executable files for Platform Manager commands. You use the executable files for the Command Central commands, and then point to the appropriate Platform Manager server using the `{--server | -s}` option.

**To execute a Platform Manager command:**

1. From the command prompt, change directory to the following location where the executable files for the Command Central commands reside:

```
Software AG_directory\CommandCentral\client\bin
```

**Note:** This step is not necessary if you have set the `CC_CLI_HOME` environment variable and included it on the `PATH` environment variable. For more information, see ["Performing Post Installation Configuration" on page 42.](#)

2. Enter the command you want to execute, using the `{--server | -s}` option to identify the Platform Manager server. For more information, see ["--server | -s" on page 77.](#)

For example, if you want to list the products that the Platform Manager server with host name `rubicon2` and port number `8092` manages, enter:

```
cc list inventory products --server http://rubicon2:8092/spm
```

**Note:** If you have set the `CC_SERVER` environment variable to the appropriate Platform Manager server, you can omit the `{--server | -s}` option.

## Getting Familiar with Using Commands

The following illustrates how you might get familiar using the Command Central and Platform Manager commands. For more information about the commands used in the examples in the following table, see ["Landscape Commands" on page 122.](#)

Step	Description	Examples
1.	Use a <code>list</code> command to view the type of information the command returns.	Execute the following command to view a list of installations. <pre>cc list landscape nodes</pre> The output includes alias names for all the installations. You can use the alias names in subsequent commands to get data for an installation, update an installation, execute actions against an installation, or delete an installation.
2.	Use a <code>get</code> or <code>list</code> command to retrieve information for a specific instance.	In this example, assume the <code>list</code> command provided information for an installation that has the alias name <code>"sag01"</code> . To retrieve information for the <code>"sag01"</code> installation, returning

Step	Description	Examples
	<p><b>Note:</b> The <code>get</code> and <code>list</code> commands are equivalent.</p>	<p>the information to an output file in XML format, execute the following command:</p> <pre>cc get landscape nodes sag01 --output info --format xml</pre>
3.	<p>Use a <code>create</code> command to create a new instance.</p> <p>You can edit the output file that a <code>get</code> command returns to specify the information for the new instance. Then you can use that file as input to the <code>create</code> command.</p>	<p>To create a new installation with alias name “new”, edit the <code>info.xml</code> file that the <code>get</code> command returned to supply the alias name, URL, and description for the new installation. Then execute the following command:</p> <pre>cc create landscape nodes --input info.xml</pre> <p><b>Note:</b> If you execute the <code>list</code> command again, the command lists the “new” installation.</p>
4.	<p>Use an <code>update</code> command to update data for an instance.</p> <p>You can use a <code>get</code> command to retrieve information for the specific instance you want to update, returning the output to a file. Then you can update the output file the <code>get</code> command returns and use that as the input to the <code>update</code> command.</p>	<p>For this example, update the “new” installation. Execute the following command to retrieve information for the “new” installation, returning the output to a file in XML format:</p> <pre>cc get landscape nodes new --output updatefile --format xml</pre> <p>Update the data in the returned “updatefile”. For example, you might specify a new description. Then execute the following command to update the installation information:</p> <pre>cc update landscape nodes new --input updatefile</pre>
5.	<p>Use an <code>exec</code> command to execute an action against an instance.</p>	<p>To generate a new ID for the “new” installation, execute the following command:</p> <pre>cc exec landscape nodes new generateNodeId</pre>
6.	<p>Use a <code>delete</code> command to remove an instance.</p>	<p>To delete the “new” installation, execute the following command:</p> <pre>cc exec landscape nodes new</pre>
<b>Note:</b>	<p>Based on the resource you are working with, all types of commands, that is <code>list</code>, <code>get</code>, <code>create</code>, <code>update</code>, <code>exec</code>, and <code>delete</code>, might not be available.</p>	

## Summary of Commands

The following table lists the commands available in the command line interface. The table also indicates whether a command is only applicable for Command Central, only applicable for Platform Manager, or can be executed on both Command Central and Platform Manager servers.

Command	Command Description	Supported on Command Central	Supported on Platform Manager
<code>cc create cfs instances</code>	Creates an instance for a Cloud Factory Services account.	X	
<code>cc delete cfs instances</code>	Deletes an instance for a Cloud Factory Services account.	X	
<code>cc get cfs instances</code>	Finds an instance for a Cloud Factory Services account.	X	
<code>cc list cfs accounts</code>	Retrieves a list of all created Cloud Factory Services accounts.	X	
<code>cc list cfs images</code>	Retrieves a list of the available images for a Cloud Factory Services account.	X	
<code>cc list cfs instances</code>	Lists the instances for a Cloud Factory Services account.	X	
<code>cc exec cfs instances</code>	Starts or stops a Cloud Factory Services account instance.	X	

Command	Command Description	Supported on Command Central	Supported on Platform Manager
<code>cc get configuration common</code>	Retrieves the schema for a specified configuration type.		X
<code>cc get configuration compare</code>	Compares a configuration type used by two or more run-time components.	X	
<code>cc create configuration data</code>	Creates a new instance of a configuration type.	X	X
<code>cc delete configuration data</code>	Deletes a configuration instance.	X	X
<code>cc get configuration data</code>	Retrieves data for a configuration instance.	X	X
<code>cc update configuration data</code>	Updates the data for a configuration instance.	X	X
<code>cc get configuration instances</code>	Retrieves information about a configuration instance.	X	X
<code>cc list configuration instances</code>	Lists the configuration instances.	X	X
<code>cc get configuration types</code>	Retrieves information for one or more configuration types.	X	X
<code>cc list configuration types</code>	Lists information for one or more configuration types.	X	X
<code>cc exec configuration validation create</code>	Validates the configuration instance data in a supplied input file.	X	X

Command	Command Description	Supported on Command Central	Supported on Platform Manager
<a href="#">cc exec configuration validation create</a>	Determines whether a configuration instance can be deleted.	X	X
<a href="#">cc exec configuration validation create</a>	Validates the configuration instance data in the supplied input file to determine whether you can use it to update a specified configuration instance.	X	X
<a href="#">cc get diagnostics logs</a>	Retrieves log entries from a log file.	X	X
<a href="#">cc get diagnostic logs export file</a>	Exports one or more log files for a specified run-time component in a zip archive file.	X	
<a href="#">cc list diagnostics logs</a>	Lists the log files that a specified run-time component supports.	X	X
<a href="#">cc create inventory components attributes</a>	Adds a new search attribute for a specified run-time component.	X	
<a href="#">cc get inventory components</a>	Retrieves information about a run-time component.	X	X
<a href="#">cc list inventory components</a>	Lists information about run-time components.	X	X
<a href="#">cc update inventory components</a>	Updates the display name and/or icon associated with a run-time component.	X	

Command	Command Description	Supported on Command Central	Supported on Platform Manager
<code>cc get inventory fixes compare</code>	Compares the fixes installed in two or more installations.	X	
<code>cc list inventory fixes</code>	Lists information about fixes that have been applied to products.	X	X
<code>cc get inventory products compare</code>	Compares the products installed in two or more installations.	X	
<code>cc get inventory products</code>	Retrieves information about a product.	X	X
<code>cc list inventory products</code>	Lists information about products.	X	X
<code>cc create instances</code>	Creates a new instance of an installed product.	X	X
<code>cc delete instances</code>	Deletes an existing instance of an installed product.	X	X
<code>cc list instances supported products</code>	Retrieves a list of products that support instance management.	X	X
<code>cc update instances</code>	Updates configuration properties of an existing instance of an installed product.	X	X
<code>cc list jobmanager jobs</code>	Lists information about long-running jobs.		X

Command	Command Description	Supported on Command Central	Supported on Platform Manager
<code>cc add landscape environments nodes</code>	Adds one or more existing installations to an environment.	X	
<code>cc create landscape environments</code>	Creates a new environment.	X	
<code>cc delete landscape environments</code>	Deletes an environment.	X	
<code>cc get landscape environments</code>	Retrieves information about an environment.	X	
<code>cc list landscape environments</code>	Lists environments in the landscape.	X	
<code>cc remove landscape environments nodes</code>	Removes one or more installations from an environment.	X	
<code>cc update landscape environments</code>	Updates the display name and/or description assigned to an environment.	X	
<code>cc create landscape nodes</code>	Adds an installation that you want to manage via Command Central.	X	
<code>cc delete landscape nodes</code>	Removes an installation from being centrally managed via Command Central.	X	
<code>cc exec landscape nodes generateNodeid</code>	Generates or regenerates a unique ID for an existing installation.	X	

Command	Command Description	Supported on Command Central	Supported on Platform Manager
<code>cc get landscape nodes</code>	Retrieves information about an installation.	X	
<code>cc list landscape nodes</code>	Lists the installations that Command Central manages.	X	
<code>cc update landscape nodes</code>	Updates the properties assigned to an installation, for example, the display name or description.	X	
<code>cc create license-tools reports snapshot</code>	Creates a license compliance snapshot report based on the currently registered nodes in a Command Central instance.	X	
<code>cc delete license-tools reports snapshot</code>	Deletes all generated license reports from Command Central.	X	
<code>cc delete license-tools reports snapshot reportid</code>	Deletes an existing license report with the specified unique report identifier.	X	
<code>cc list license-tools reports snapshot</code>	Lists all license reports available on the Command Central server.	X	
<code>cc get license-tools reports snapshot reportid</code>	Obtains information about a license report with the specified unique report identifier.	X	
<code>cc get license-tools reports snapshot output PDF</code>	Generates a PDF file for an existing license report.	X	

Command	Command Description	Supported on Command Central	Supported on Platform Manager
<code>cc get license-tools reports snapshot output XML</code>	Generates an existing license report in XML format.	X	
<code>cc create license-tools reports installation</code>	Creates a license report of the current landscape based on a license manifest file.	X	
<code>cc delete license-tools reports installation reportid</code>	Deletes an existing license report by license manifest with the specified unique report identifier.	X	
<code>cc delete license-tools reports installation</code>	Deletes all generated reports by license manifest.	X	
<code>cc get license-tools reports installation output PDF</code>	Generates a PDF file for an existing license report by license manifest.	X	
<code>cc get license-tools reports installation output XML</code>	Generates an existing license report by license manifest in XML format.	X	
<code>cc get license-tools reports installation disclaimer</code>	Displays the disclaimer text that is valid for the reports by license manifest.	X	
<code>cc list license-tools reports installation</code>	Lists all available reports by license manifest.	X	
<code>cc list license-tools reports</code>	Lists all available reports, both by license key and by license manifest	X	
<code>cc list license-tools metering products</code>	Lists the names and codes of all products for	X	

Command	Command Description	Supported on Command Central	Supported on Platform Manager
	which license reports are available.		
<code>cc add license-tools manifests</code>	Adds a license manifest file with the specified manifest alias.	X	
<code>cc delete license-tools manifests</code>	Deletes all available license manifest files or a license manifest with the specified manifest alias.	X	
<code>cc get license-tools manifests output xml</code>	Generates an existing license manifest file in XML format.	X	
<code>cc list license-tools manifests</code>	Lists all available license manifests.	X	
<code>cc exec lifecycle</code>	Executes an action to start, stop, pause, and/or resume run-time components.	X	X
<code>cc get monitoring</code>	Retrieves the run-time statuses, run-time states, or states of run-time components.	X	X
<code>cc list monitoring alerts</code>	Lists the alerts for a specified run-time component.	X	
<code>cc create provisioning bootstrap installers</code>	Command Central creates a bootstrap installer image from a local or remote master product repository, configured with valid user credentials, and registers the bootstrap	X	

Command	Command Description	Supported on Command Central	Supported on Platform Manager
	installer image as an image product repository.		
<code>cc list provisioning bootstrap installers</code>	Retrieves a list of available bootstrap installer images.	X	
<code>cc exec provisioning bootstrap nodes</code>	Installs a bootstrap installer image on a local or remote machine and registers the new Platform Manager installation under a specified node alias.	X	
<code>cc exec provisioning products install</code>	Installs products from a product repository	X	
<code>cc exec provisioning fixes install</code>	Installs fixes from a fix repository.	X	
<code>cc exec provisioning uninstall</code>	Uninstalls products or fixes.	X	
<code>cc add repository fixes</code>	Adds a product or fix repository.	X	
<code>cc delete repository</code>	Deletes a registered product or fix repository.	X	
<code>cc delete repositories</code>	Deletes all registered product or fix repositories.	X	
<code>cc exec repository discover</code>	Finds product and fix repositories for the specified host, name, and port and adds the discovered repositories to Command Central.	X	X

Command	Command Description	Supported on Command Central	Supported on Platform Manager
<code>cc exec repository register</code>	Copies a product or fix repository, including its image file, to a new Platform Manager node.	X	
<code>cc list repository</code>	Lists registered product or fix repositories.	X	
<code>cc list repository discover</code>	Finds product or fix repositories for the specified host, name, and port, but does not add the discovered repositories to Command Central.	X	
<code>cc list repository fixes content</code>	Lists the fixes available in a fix repository.	X	
<code>cc list repository fixes dependencies</code>	Checks the dependencies for a fix.	X	
<code>cc list repository fixes readme</code>	Retrieves the readme for a fix.	X	
<code>cc list repository platforms</code>	Lists all supported operating systems on the specified Platform Manager node.	X	X
<code>cc list repository products content</code>	Lists the products available in a product repository.	X	
<code>cc list repository products dependencies</code>	Checks the dependencies for a product.	X	
<code>cc update repository</code>	Updates a repository using data from an XML file.	X	

Command	Command Description	Supported on Command Central	Supported on Platform Manager
<a href="#">cc update repository details</a>	Updates a repository description and location.	X	
<a href="#">cc list resources icons</a>	Lists information about the installed icons.	X	
<a href="#">cc add security credentials</a>	Adds security credentials.	X	
<a href="#">cc delete security credentials</a>	Deletes security credentials.	X	
<a href="#">cc get security credentials</a>	Retrieves security credentials.	X	
<a href="#">cc create templates snapshot</a>	Creates a new template for an existing managed installation.	X	X
<a href="#">cc delete templates snapshot</a>	Removes a template from an installation.	X	
<a href="#">cc exec templates snapshot apply</a>	Applies a template registered and available in a managed installation.	X	X
<a href="#">cc export templates snapshot</a>	Exports an existing template.	X	X
<a href="#">cc list templates snapshot</a>	Lists all templates available in a landscape.	X	X
<a href="#">cc exec templates snapshot import</a>	Registers an exported template.	X	X
<a href="#">cc delete templates composite</a>	Deletes a composite template with the specified alias.	X	

Command	Command Description	Supported on Command Central	Supported on Platform Manager
<code>cc exec templates composite apply</code>	Applies a composite template.	X	
<code>cc exec templates composite import</code>	Registers an existing composite template.	X	
<code>cc list templates composite</code>	Retrieves a list of composite templates available in a landscape.	X	

## Return Codes from Command Execution

The following table lists the return codes that can result from executing a Command Central or Platform Manager command.

Return Code	Description
0	Indicates the execution of the command was successful. A command returns 0 when the HTTP response code is less than 400.
1	Indicates that the command syntax is not valid.
10	Indicates the output that a command returned does not match the expected values specified with the <code>{--expected-values   -e}</code> option.
<i>response-code</i>	Indicates the command execution resulted in an error. The return code is the HTTP response code. A command uses this return code when the HTTP response code is greater than or equal to 400.

**Note:** On Unix, return codes greater than 256 are considered "Out of range exit values." As a result, when executing the commands on a client machine that runs on Unix, if the HTTP response code is greater than 256, the return code is *response-code* modulo 256.

## Options for the Commands

### Common Options

The following are options that the Command Line Interface (CLI) supports. To determine the options that a specific command allows, see the documentation for that command.

- `--accept | -a` on page 59
- `--check-every | -c` on page 60
- `--configuration-file` on page 61
- `--debug | -d` on page 62
- `--error | r` on page 63
- `--expected-values | -e` on page 64
- `--force` on page 66
- `--format | -f` on page 67
- `--input | -i` on page 70
- `--input-format | -m` on page 72
- `--log | -l` on page 73
- `--media-type | -m` on page 74
- `--output | -o` on page 75
- `--output-format | -f` on page 76
- `--password | -p` on page 76
- `--quiet | -q` on page 77
- `--server | -s` on page 77
- `--ssl-truststore-file` on page 78
- `--ssl-trust-all-hosts` on page 79
- `--ssl-truststore-password` on page 79
- `--username | -u` on page 80
- `--wait | -w` on page 80

**Note:** When you use both a deprecated option and the new option that replaces it in the same command, the new option overrides the value of the deprecated option.

### `--accept | -a`

Deprecated. Use `--output-format | -f` in place of `--accept | -a`. When you use `--accept | -a`, Command Central executes the command with a warning.

Specifies the format you want the command to use for the data it returns. Use the `{--accept | -a}` option to specify a content type that the command supplies on the HTTP Accept request header that it sends to Command Central or Platform Manager.

## Syntax

```
{--accept | -a} content_type
```

## Arguments and Options

Argument	Description
<code>content_type</code>	<p>Specifies a well-formed content type that indicates the format you want the command to use for the output. The following lists some examples:</p> <ul style="list-style-type: none"> <li>■ application/xml</li> <li>■ application/json</li> <li>■ text/plain</li> <li>■ text/tab-separated-values</li> <li>■ text/csv</li> </ul>

## Usage Notes

- If you specify the `{--input | -i}` option, the command ignores the `{--accept | -a}` option and sets the request content type based on the file extension of the input file. For more information, see "[--input | -i](#)" on page 70.
- Use the `{--accept | -a}` option as an alternative to the `{--format | -f}` option. Both options set the request content type.
- If you specify both the `{--accept | -a}` option and the `{--format | -f}` option, the command uses the content type you specify with the `{--accept | -a}` option and ignores the `{--format | -f}` option.
- By default, output is written to the console. If you want the output written to a file, use the `{--output | -o}` option. For more information, see "[--output | -o](#)" on page 75.

## Examples

- To have a command return data in JavaScript Object Notation format:
 

```
--accept application/json
```
- To have a command return data in csv format:
 

```
--accept text/csv
```

## **--check-every | -c**

Specifies how often (in seconds) to check whether a long-running operation has returned the expected values. Use in conjunction with the [--expected-values | -e](#) and [--wait | -w](#) options.

## Syntax

```
{--check-every | -c} seconds
```

## Arguments

Argument	Description
<i>seconds</i>	<p>Specifies the number of seconds the command waits before checking for expected output specified by the <code>{--expected-values   -e}</code> option.</p> <p>If you omit the <code>{--check-every   -c}</code> option, the command uses the value of the <code>CC_CHECK_EVERY</code> environment variable. If the <code>CC_CHECK_EVERY</code> environment variable is not set, the command uses 15 seconds.</p>

## Usage Notes

- The `{--check-every | -c}` option is only needed when you specify the `{--expected-values | -e}` option.
- The command is continually executed every `{--check-every | -c}` seconds until the command either returns the expected values or times out because the seconds specified by the `{--wait | -w}` option have elapsed.
- If the time specified by the `{--wait | -w}` option elapses before the expected results are returned, the command fails.
- The use of the `{--expected-values | -e}`, `{--wait | -w}`, and `{--check-every | -c}` options is helpful with commands that perform actions that might take several seconds or minutes to complete. Depending on your use case, these options might be helpful with any command. However, they are most helpful with the `lifecycle` and `monitoring` commands because they allow you to reliably execute the commands.

## Example

To have a command check every 30 seconds for the expected results:

```
--check-every 30
```

**Note:** To see an example that uses all of the `{--expected-values | -e}`, `{--wait | -w}`, and `{--check-every | -c}` options, see "[--expected-values | -e](#)" on page 64.

## --configuration-file

Specifies the location of the configuration file that contains a list of configuration properties, such as SSL, server, username, and password settings.

## Syntax

`--configuration-file path`

## Arguments

Argument	Description
<i>path</i>	<p>Specifies the fully qualified path to the location of the configuration properties file.</p> <p>The default Command Central <code>cc.properties</code> file is located in the <code>Software AG_directory\CommandCentral\client\conf</code> directory.</p> <p><b>Note:</b> You can specify a relative path to the current execution directory of the <code>cc</code> command if you have set the <code>CC_CLI_HOME</code> and <code>PATH</code> environment variables.</p>

## Usage Notes

- When you include both the `--configuration-file` and the `-ssl-truststore-password` options, Command Central uses the password specified in the `-ssl-truststore-password` option.
- If you do not want to specify the SSL truststore options for each command execution, you can include the `--ssl-truststore-file`, `--ssl-truststore-password`, and `--ssl-trust-all-hosts` options in a custom `cc.properties` file and specify the path to the location of that file in the `--configuration-file` option. For more information about creating a custom configuration properties file, see ["Configuring SSL Using Configuration Properties Files"](#) on page 166.

## --debug | -d

Specifies you want the command to return extra information that you can use for debugging issues, in addition to the returning service output. The extra information includes:

- HTTP service request
- URL of the Command Central or Platform Manager server to which the request was sent
- Request content type
- Accept header for the request
- HTTP response code from the request
- Response content type
- Response content length

## Syntax

```
{--debug | -d}
```

## Arguments

None

## Usage Notes

- If you specify both `{--debug | -d}` and `{--quiet | -q}`, the command ignores the `{--quiet | -q}` option and uses the `{--debug | -d}` option to display the additional debug information.

## Example

The following shows sample output that uses the `--debug` option.

```
cc list landscape nodes --debug
```

```

Debug information | Request: GET http://localhost:8090/occe/landscape/nodes
                  | Host: localhost:8090
                  | Content-Type: text/tab-separated-
                  | values;text/plain,application/xml;q=0.9,*/*;q=0.8
                  | Accept: text/tab-separated-values;text/plain,application/xml;q=0.9,*/*;q=0.8
                  | Response: 200 OK
                  | Content-Type: text/tab-separated-values
                  | Content-Length: 89
Service Output   | Alias  Name                Status  Url  Host  Url Port
                  | node125 Name of node node125  ONLINE localhost 8202 ]]]]

```

## --error | r

Specifies a file where you want a command to write the output if the command results in an error. If you do not specify the `{--error | -r}` option, the command writes the output to the console.

## Syntax

```
{--error | -r} file
```

## Arguments

Argument	Description
<i>file</i>	Specifies the file where you want the error output written. If the file you specify does not exist, the command creates it. You can specify: <ul style="list-style-type: none"> <li>■ Absolute directory path and filename.</li> <li>■ Relative directory path and filename. The path is relative from where you initiated the command.</li> </ul>

Argument	Description
	<ul style="list-style-type: none"> <li>■ Filename of a file in the same directory where you initiated the command.</li> </ul>

### Usage Notes

- If the file you specify with the `{--error | -r}` option already exists, the command overwrites the existing file with the new service results.
- If a command encounters an error, to help resolve errors, you can execute the command again using the `{--debug | -d}` option to display additional information about the actual request and response.
- You can use the `{--error | r}` option to direct error results to a specific location, for example, if you want to use automated tools to review output.
- If a command executes successfully, the command writes the output to the location specified by the `{--output | -o}` option or the console if the `{--output | -o}` option is not specified.

### Examples

- To write error output to a file named “errors.xml” in the directory `c:\outputs`:  

```
--error c:\outputs\errors.xml
```
- To write error output to a file named “errors.json” in the `\outputs` directory relative to where you initiate the command:  

```
--error outputs\errors.json
```
- To write output to a file named “errors” in the same directory from where you initiate the command:  

```
--error errors
```

In this example, the command determines the file extension based on the request content type.

## --expected-values | -e

Specifies the expected values for which to wait before a command completes. Use in conjunction with the `--check-every | -c` and `--wait | -w` options.

When you use the `{--expected | -e}`, `{--check-every | -c}`, and `{--wait | -w}` options, each `{--check-every | -c}` seconds the command is executed and the results are examined for the values specified with the `{--expected | -e}` option. The command is successful if the command returns the expected values within the wait time. The command fails if the command does not return the expected values within the wait time.

## Syntax

```
{--expected-values | -e} values
```

## Arguments

Argument	Description
<i>values</i>	<p>Specifies the values that must be present in the output for a command to complete. Use a comma to separate each value. Always place quotes around the value, for example:</p> <pre>--expected-values "STOPPED,ONLINE"</pre> <p>The values can include the wildcard character (*) that represents any number of characters. For example, <code>RUNN*</code>, <code>*NLINE</code>, <code>OFF*NE</code>.</p>

## Operators

The command supports the following operators listed in the table by operator precedence:

Operator	Description
!	Negates the argument to the right, for example <code>!RUNNING</code> returns all responses that do not contain the text "RUNNING".
,	<p>Specifies the sequence in which the values will get executed. For example, if you specify <code>STOPPED,UNKNOWN,ONLINE</code>, the command first checks for the <code>STOPPED</code> run-time status. After <code>STOPPED</code>, the command checks for the <code>UNKNOWN</code> status, and after that it checks for <code>ONLINE</code>. If any of the run-time statuses do not occur before the command times out, the command returns an error indicating the missing statuses.</p> <p>When you specify multiple values, the command checks for <i>all</i> values and completes successfully only if <i>all</i> of them are present.</p>
&	Indicates that the value includes a logical AND operator. For example, <code>RUNNING&amp;RESPONSIVE</code> checks if the output contains both <code>RUNNING</code> and <code>RESPONSIVE</code> .
	<p>Indicates that the value includes a logical OR operator, for example the following command checks if the output contains either <code>DONE</code> or <code>WARNING</code>:</p> <pre>--expected "DONE WARNING"</pre>

## Usage Notes

- You can define the expected values as a complex expression that contains more than one operator. The order of the operations is controlled by using brackets, for example: "`!RU*NG|OFFLINE&!ONLINE`" and "`(!RU*NG)|(OFFLINE&(!ONLINE))`" follow the same order of operations, but "`(!(RU*NG|OFFLINE))&(!ONLINE)`" follows a different operator precedence.
- If a command is executed from a shell on Windows or UNIX operating systems, the exclamation mark is considered a special character. Use the following escape characters to escape (!):
  - (^) on Windows, for example "`^!RUNNING`"
  - (\) on UNIX/Linux, for example "`\!RUNNING`"
- The use of the `{--expected-values | -e}`, `{--wait | -w}`, and `{--check-every | -c}` options is helpful with commands that perform actions that might take several seconds or minutes to complete. Depending on your use case, these options might be helpful with any command. However, they are most helpful with the `lifecycle` and `monitoring` commands because they allow you to reliably execute the commands.
- If you specify the `{--expected-values | -e}` option, but omit the `{--check-every | -c}` option, the command uses the value from the `CC_CHECK_EVERY` environment variable. If the `CC_CHECK_EVERY` environment variable is not set, the command uses 15 seconds.
- If you specify the `{--expected-values | -e}` option, but omit the `{--wait | -w}` option, the command uses the value from the `CC_WAIT` environment variable. If the `CC_WAIT` environment variable is not set, the command uses 15 seconds.

## Example

To wait 180 seconds for a command to return the value "STOPPED", then "ONLINE", checking every 30 seconds for the expected results:

```
--expected-values "STOPPED,ONLINE" --wait 180 --check-every 30
```

## --force

Forces the execution of a delete command without prompting for confirmation of the requested operation.

### Syntax

```
--force
```

### Arguments

None.

## Usage Note

When you omit the `--force` option, the delete command prompts you to confirm the requested operation.

## --format | -f

Deprecated. Use `--output-format | -f` in place of `--format | -f`. When you use `--format | -f`, Command Central executes the command with a warning.

Specifies the format you want a command to use for the data it returns. Command Central and Platform Manager support the following formats:

- Tab-separated values (tsv)
- Plain text (txt)
- XML (xml)
- Comma-separated values (csv)
- JavaScript Object Notation (json)

Although Command Central and Platform Manager support these formats, a specific command might only support a subset of the formats. Refer to the documentation for a specific command to determine the exact formats that it supports.

## Syntax

```
{--format | -f} {tsv args | text | xml | csv args | json}
```

## Arguments

Argument	Description
<code>tsv args</code>	Specifies you want the output in tab-separated values format. For more information about the arguments you can specify, see <a href="#">"Arguments for Tab-Separated Values (tsv) and Comma-Separated Values (csv)"</a> on page 68.
<code>text</code>	Specifies you want the output in plain text format.
<code>xml</code>	Specifies you want the output in XML format.
<code>csv args</code>	Specifies you want the output in comma-separated values format. For more information about the arguments you can specify, see <a href="#">"Arguments for Tab-Separated Values (tsv) and Comma-Separated Values (csv)"</a> on page 68.
<code>json</code>	Specifies you want the output in JavaScript Object Notation format.

## Usage Notes

- Use the `{--format | -f}` option as an alternative to the `{--accept | -a}` option. Both options set the request content type. For more information, see ["--accept | -a" on page 59](#).
- If you specify both the `{--format | -f}` option and the `{--accept | -a}` option, the command uses the content type you specify with the `{--accept | -a}` option and ignores the `{--format | -f}` option.
- If you specify the `{--input | -i}` option, the command ignores the `{--format | -f}` option and sets the request content type based on the file extension of the input file. For more information, see ["--input | -i" on page 70](#).
- By default, output is written to the console. If you want the output written to a file, use the `{--output | -o}` option. For more information, see ["--output | -o" on page 75](#).
- The following describes the typical default that a command uses if you do not specify the `{--format | -f}` option:
  - If you execute the command from the command line, a batch script, or a shell script, the default format is tab-separated values (tsv) format.
  - If you execute the command from an Ant script, the default format is XML format.

To determine the default for a command that does not support tab-separated values (tsv) or XML format, refer to the documentation for that command.
- If a command supports either the `tsv` or `csv` format, you can restrict the fields the command returns to only those fields you specify. For more information, see ["Arguments for Tab-Separated Values \(tsv\) and Comma-Separated Values \(csv\)" on page 68](#).

## Examples

- To have a command return data in csv format without headers:
 

```
--format csv includeHeaders=false
```
- To have a command return data in xml format:
 

```
--format xml
```

## Arguments for Tab-Separated Values (tsv) and Comma-Separated Values (csv)

When you specify the `tsv` or `csv` with the `{--format | -f}` option, you can specify additional arguments to customize the output.

### Syntax

```
{--format | -f} {tsv | csv} [includeHeaders={labels | properties | none}]
[properties=keys]
```

## Argument Descriptions

Argument	Description
<pre>[includeHeaders={labels   properties   none}]</pre>	<p>Specifies whether you want the output to include a header line. Specify one of the following:</p> <ul style="list-style-type: none"> <li>■ <code>labels</code> to include a header line containing the display names for each field. For example “Product Version” might be a display name if the output includes the version of a product. This is the default.</li> <li>■ <code>properties</code> to include a header line containing the property key name for each field. For example “product.version” might be the display name if the output includes the version of a product.</li> <li>■ <code>none</code> to omit headers from the output.</li> </ul>
<pre>[properties=keys]</pre>	<p>Identifies the keys associated with the information you want included in the output. For example, if you want the output to only include the product version, specify <code>properties=product.version</code>.</p> <p>To specify multiple keys, separate each with a comma. For example, if you want alias names and descriptions in the output, you might specify <code>properties=alias,description</code>.</p> <p>Use <code>properties=*</code> to include all information. If you omit the <code>properties</code> argument, the command returns a default set of fields.</p>

## Usage Notes

- To determine the keys you can specify with the `properties` argument, execute a `get` or `list` command and specify `includeHeaders=properties properties=*` so that the output displays a header line that shows the keys for all the possible properties.

For example, you might want to use the `cc list landscape nodes` command to retrieve the list of alias names and descriptions for installations. First, execute the `cc list landscape nodes` command with `--format csv includeHeaders=properties properties=*` to determine that the key for the alias name is `alias` and the key for the description is `description`. You can then

execute `cc list landscape nodes` with `--format csv includeHeaders=name properties=alias,description`.

## --input | -i

Identifies a file that contains the input data for a `create`, `add`, `update`, or `exec` command.

For example, when using the `cc create landscape nodes` command to add a new installation that you want Command Central to manage, you are required to provide an alias name for the installation and the URL for the installation. You can provide this information on the command line using command line arguments, or you can use the `{--input | -i}` option to specify this data in an input file. For some commands, the item you are creating, adding, or updating requires more data than is practical to supply on the command line, and as a result, the `{--input | -i}` option might be required to supply the data for the command.

### Syntax

```
{--input | -i} filename { .xml | .json | .properties }
```

### Arguments

Argument	Description
<code>filename { .xml   .json   .properties }</code>	<p>Specifies the file that contains the input data. The input file can be:</p> <ul style="list-style-type: none"> <li>■ An <code>.xml</code> file containing input data in XML format</li> <li>■ A <code>.json</code> file containing input data in JavaScript Object Notation format</li> <li>■ A <code>.properties</code> file containing input data in key/value pairs format.</li> </ul> <p>When identifying the input file, you can specify:</p> <ul style="list-style-type: none"> <li>■ Absolute directory path and filename.</li> <li>■ Relative directory path and filename. The path is relative from where you initiated the command.</li> <li>■ Filename of a file in the same directory where you initiated the command.</li> </ul>

### Usage Notes

- The use of an input file for data is helpful when:
  - You are scripting commands, for example, using an Ant script.

- You are executing a command with complicated input parameters where it is easier to specify them in a file in XML or json format rather than specifying them on the command line.
- You want to create templates for adding items such as installations, configurations, etc.
- A command always sets the request content type based on the file extension of the input file if the `{--input | -i}` option is specified. This is true even if you specify another option that affects the request content type, that is, the `{--accept | -a}` or `{--format | -f}` option.

The following lists the request content type that a command uses based on the file extension of the input file:

<u>File extension</u>	<u>Request content type</u>
.xml	application/xml
.json	application/json
.properties	application/x-www-form-urlencoded

- The `{--input | -i}` option is supported for POST and PUT requests, that is for `create`, `add`, `exec` and `update` commands. It is not supported for GET and DELETE requests, that is `get`, `list`, `delete`, or `remove` commands.
- The input file contains data that a command requires for creating an item, for updating an item, or for the execution of an operation. You must supply a file in the format that the command expects, using specific element names and/or tags. For example, when using the `cc create configuration data` command to create an instance of a COMMON-PORTS configuration type, to supply the port number in an XML file, you might include the element `<Number>5555</Number>` as part of the XML file.

To determine the format to use for an input file, execute a `get` or `list` command to retrieve a similar item. On the `get` or `list` command, if you use the `{--output | -o}` option to write the output to a file, you can then update the returned output file and specify it with the `{--input | -i}` option as an input data file.

For example, if you want to use the `cc create configuration data` command to create a COMMON-PORTS configuration instance, first use the `cc get configuration data` command to retrieve an existing COMMON-PORTS instance to learn the format to use for the input data file.

- If you specify input data both on the command line and use the `{--input | -i}` option to specify data in an input file, the command uses the data that you specify in the input file and ignores the data you specify on the command line.

## Examples

- To use the input file `input.xml` in the directory `c:\templates`:  

```
--input c:\templates\input.xml
```
- To use the input file `input.xml` in the `\templates` directory relative to where you initiate the command:  

```
--input templates\input.xml
```
- To use the input file `input.xml` that resides in the same directory from where you initiate the command:  

```
--input input.xml
```

## --input-format | -m

Specifies the content type of the input data for a command. You can specify the same values for `--input-format | -m` and `--output-format | -f`.

### Syntax

```
--input-format | -m content-type
```

### Arguments

Argument	Description
<i>content-type</i>	<p>Specifies a well-formed content type that indicates the format you want the command to use for the input. You can specify the short or full versions of the media type. The following lists some examples:</p> <ul style="list-style-type: none"> <li>■ <code>application/xml   xml</code></li> <li>■ <code>application/json   json</code></li> <li>■ <code>text/plain   text</code></li> <li>■ <code>text/tab-separated-values   tsv</code></li> <li>■ <code>text/csv   csv</code></li> </ul> <p>The default value is taken from the input file extension if the extension matches the short version of a supported media type. If the input file extension does not match the short version of a supported media type, the default is <code>text/plain</code>.</p> <p>For the content types that a command supports, see the documentation for a specific command.</p>

## Example

To specify the input data is content type application/xml:

```
--input-format application/xml
```

## --log | -l

Specifies a file where you want to log the outcome from the execution of the command, whether the command completes successfully or encounters errors. If you do not specify the `{--log | -l}` option, the command logs this error information to the console.

The logged results include:

- Service output
- Errors that occur while interpreting the command

**Note:** If the error occurs while the initializing the command, the error is written to the console rather than the file specified with the `{--log | -l}` option

- Debug information if the `{--debug | -d}` option is specified on the command

## Syntax

```
--log | -l file
```

## Arguments

Argument	Description
<i>file</i>	Specifies the log file where you want the errors written. If the file you specify does not exist, the command creates it. You can specify: <ul style="list-style-type: none"> <li>■ Absolute directory path and filename.</li> <li>■ Relative directory path and filename. The path is relative from where you initiated the command.</li> <li>■ Filename of a file in the same directory where you initiated the command.</li> </ul>

## Usage Notes

- If you use the `{--output | -o}` option with the `{--log | -l}` option and the command completes successfully, the command writes the results to the output file *and* logs the outcome to the log file.

- If you use the `{--error | -r}` option with the `{--log | -l}` option and the command encounters an error, the command writes the error results to the error file *and* logs the outcome to the log file.
- If a command uses the `{--debug | -d}` command, the debug information is also written to the log file.
- If the file you specify with the `{--log | -l}` option already exists, the command appends the new service results to the file.
- The error information includes a timestamp. Using this option for commands generates a history of the command execution and actions.

### Examples

- To log information to a file named “logfile.xml” in the directory `c:\outputs`:  

```
--log c:\outputs\logfile.xml
```
- To log information to a file named “logfile.json” in the `\outputs` directory relative to where you initiate the command:  

```
--log outputs\logfile.json
```
- To log information to a file named “logfile” in the same directory from where you initiate the command:  

```
--log logfile
```

## --media-type | -m

Deprecated. Use `--input-format | -m` in place of `--media-type | -m`. When you use `--media-type | -m`, Command Central executes the command with a warning.

Specifies the content type of the input data.

### Syntax

```
--media-type | -m content-type
```

### Arguments

Argument	Description
<i>content-type</i>	Specifies the content type. Refer to the documentation for a specific command to determine the content types that a command supports.

### Example

To specify the input data is content type `application/xml`:

```
--media-type application/xml
```

## --output | -o

Specifies a file where you want a command to write the output if the command executes successfully. If you do not specify the `{--output | -o}` option, the command writes the output to the console.

### Syntax

```
--output | -o file
```

### Arguments

Argument	Description
<i>file</i>	<p>Specifies the file where you want the output written. If the file you specify does not exist, the command creates it. You can specify:</p> <ul style="list-style-type: none"> <li>■ Absolute directory path and filename.</li> <li>■ Relative directory path and filename. The path is relative from where you initiated the command.</li> <li>■ Filename of a file in the same directory where you initiated the command.</li> </ul>

### Usage Notes

- If the file you specify with the `{--output | -o}` option already exists, the command overwrites the existing file with the new service results.
- If a command results in an error, the command writes the error output to the location specified by the `{--error | -r}` option or the console if the `{--error | -r}` option is not specified.

### Examples

- To write output to a file named "results.xml" in the directory `c:\outputs`:
 

```
--output c:\outputs\results.xml
```
- To write output to a file named "results.json" in the `\outputs` directory relative to where you initiate the command:
 

```
--output outputs\results.json
```
- To write output to a file named "results" in the same directory from where you initiate the command:
 

```
--output results
```

In this example, the command determines the file extension based on the request content type.

## **--output-format | -f**

Specifies the content type of the output data for a command. You can specify the same values for `--input-format | -m` and `--output-format | -f`.

### **Syntax**

```
--output-format | -f content-type
```

### **Arguments**

Argument	Description
<i>content-type</i>	<p>Specifies a well-formed content type that indicates the format you want the command to use for the output. You can specify the short or full versions of the media type. The following lists some examples:</p> <ul style="list-style-type: none"> <li>■ <code>application/xml   xml</code></li> <li>■ <code>application/json   json</code></li> <li>■ <code>text/plain   text</code></li> <li>■ <code>text/tab-separated-values   tsv</code></li> <li>■ <code>text/csv   csv</code></li> </ul> <p>The default value is taken from the output file extension if the extension matches the short version of a supported media type. If the output file extension does not match the short version of a supported media type, the default is <code>text/tab-separated-values</code>.</p> <p>For the content types that a command supports, see the documentation for a specific command.</p>

### **Example**

To specify the output data is content type `application/xml`:

```
--output-format application/xml
```

## **--password | -p**

Specifies the password for the user executing the command.

### **Syntax**

```
{--password | -p} password
```

### Usage Notes

- If you omit the `{--password | -p}` option, the command uses the value from the `CC_PASSWORD` environment variable. If the `CC_PASSWORD` environment variable is not set, the command prompts the user for the password.

### Example

To specify the password “secret”:

```
--password secret
```

## --quiet | -q

Specifies that you want the command to return only service output with no additional information.

### Syntax

```
{--quiet | -q}
```

### Arguments

None.

### Usage Notes

- If you specify both `{--debug | -d}` and `{--quiet | -q}`, the command ignores the `{--quiet | -q}` option and uses the `{--debug | -d}` option to display additional debug information.

## --server | -s

Identifies the Command Central or Platform Manager server on which to execute a command.

### Syntax

```
{--server | -s} url
```

### Arguments

Argument	Description
<i>url</i>	<p>Identifies the URL of a Command Central or Platform Manager server on which to execute the command.</p> <ul style="list-style-type: none"> <li>■ When specifying the URL of a Command Central server, if you omit: <ul style="list-style-type: none"> <li>■ Protocol, the command uses “https://”.</li> </ul> </li> </ul>

Argument	Description
	<ul style="list-style-type: none"> <li>■ Port number, the command uses “8090”, which is the default port for a Command Central server.</li> <li>■ Context, the command uses “cce”.</li> <li>■ You must always provide the full URL, including the protocol, when identifying a Platform Manager server.</li> </ul>

### Usage Notes

- If you omit the `{--server | -s}` option, the command uses the value from the `CC_SERVER` environment variable. If the `CC_SERVER` environment variable is not set, the command executes on `localhost:8090`.
- If you want to execute a command on a Platform Manager server, either specify the `{--server | -s}` option on the command or ensure the `CC_SERVER` environment variable specifies a Platform Manager server.

### Examples

- To execute a command on the Command Central server with host name `rubicon` and port number `8090` using the `http` protocol:

```
--server rubicon
--server rubicon:8090
--server http://rubicon:8090
--server http://rubicon:8090/cce
```

- To execute a command on the Platform Manager server with host name `rubicon2` and port number `8092` using the `http` protocol:

```
--server http://rubicon2:8092/spm
```

## --ssl-truststore-file

Specifies the location of the truststore file.

### Syntax

```
--ssl-truststore-file=path
```

### Arguments

Argument	Description
<i>path</i>	<p>Specifies the fully qualified path to the truststore location.</p> <p>The default Command Central truststore with name <code>demo-truststore.jks</code> is located in the <i>Software AG_directory</i>\CommandCentral\client\conf directory.</p>

Argument	Description
	<p><b>Note:</b> You can specify a relative path to the current execution directory of the <code>cc</code> command if you have set the <code>CC_CLI_HOME</code> and <code>PATH</code> environment variables.</p>

### Usage Notes

If you do not include the `--ssl-truststore-file` option, the command fails to execute.

### Example

To execute the command with the Command Central default truststore:

```
--ssl-truststore-file=Software AG_directory\CommandCentral\client\
conf\demo-truststore.jks
```

## --ssl-trust-all-hosts

Specifies whether to trust all hosts. When the option is included in the command, Command Central does not verify the name of the server host.

### Syntax

```
--ssl-trust-all-hosts
```

### Arguments

None.

### Usage Notes

- The default Command Central and Platform Manager keystore with name `demo-keystore.jks` contains a signed CA certificate. However, the generated CA certificate does not contain the fully qualified name of the server host required by the client to trust the server. When the `ssl-trust-all-hosts` option is included, Command Central does not verify the name of the server host.
- If you do not include the `--ssl-trust-all-hosts` option, Command Central will attempt to verify the server host name, and the client will not trust a server certificate without a fully qualified server host name. When the server host name matches the host name in the signed CA certificate, the option is ignored.

## --ssl-truststore-password

Specifies the password for the truststore.

### Syntax

```
--ssl-truststore-password=password
```

### Usage Notes

- When you do not provide a truststore password or you specify the wrong password, the command fails.
- When you include both the `--configuration-file` and the `-ssl-truststore-password` options in the command, Command Central uses the password specified in the `-ssl-truststore-password` option.

### --username | -u

Specifies the user who is executing a command. The specified user must have the proper authorization to execute the command.

#### Syntax

```
{--username | -u} user_name
```

#### Arguments

Argument	Description
<code>user_name</code>	Specifies the user name of the user executing the command.

### Usage Notes

- If you omit the `{--username | -u}` option, the command uses the value from the `CC_USERNAME` environment variable. If the `CC_USERNAME` environment variable is not set, the command uses "Administrator".
- Use the `{--password | -p}` option to specify the user's password. If you omit the `{--password | -p}` from the command line, the command will prompt you for the password. For more information, see "[--password | -p](#)" on page 76

### Example

To execute the command as the user with user name admin02:

```
--username admin02
```

### --wait | -w

Specifies how many seconds to wait for a long-running operation to return the expected values. Use in conjunction with the `--expected-values | -e` and `--check-every | -c` options.

#### Syntax

```
{--wait | -w} seconds
```

## Arguments

Argument	Description
<code>seconds</code>	<p>Specifies the number of seconds the command waits for the expected output specified by the <code>{--expected-values   -e}</code> option before completing.</p> <p>The default is the value of the <code>CC_WAIT</code> environment variable. If the <code>CC_WAIT</code> environment variable is not set, the command uses 120 seconds.</p>

## Usage Notes

- The `{--wait | -w}` option is only needed when you specify the `{--expected-values | -e}` option.
- If the time specified by the `{--wait | -w}` option elapses before the expected results are returned, the command fails.
- The use of the `{--expected-values | -e}`, `{--wait | -w}`, and `{--check-every | -c}` options is helpful with commands that perform actions that might take several seconds or minutes to complete. Depending on your use case, these options might be helpful with any command. However, they are most helpful with the `lifecycle` and `monitoring` commands because they allow you to reliably execute the commands.

## Example

To have a command wait 180 seconds for the expected results:

```
--wait 180
```

**Note:** To see an example that uses all of the `{--expected-values | e}`, `{--wait | -w}`, and `{--check-every | -c}` options, see "[--expected-values | -e](#)" on page 64.

## Invoking Commands from Scripts

### Creating Shell Scripts that Execute Commands

On Windows, execute `cc` commands within a `.bat` file execute using `call` statements. The following is an example of a script that might be in a file named `get-products-inventory.bat`:

```
@echo off
echo getting products inventory
call cc list inventory products
```

## Creating Ant Scripts that Execute Commands

You can create Apache Ant scripts that execute Command Central and Platform Manager command line interface commands.

When creating your Ant script, you must:

1. Use the following fragment to declare `cc` Ant tasks:

```
<property environment="os" />
<property="cc.home" value="${os.CC_CLI_HOME}" />

<taskdef resource="com/softwareag/platform/management/client/ant/antlib.xml"
  <classpath>
    <fileset dir="${cc.home}/lib">
      <include name="*.jar" />
    </fileset>
  </classpath>
</taskdef>
```

2. Create one or more targets that use the `ccsetup` and `cc` tasks. The following shows a sample:

```
<target name="execute-commands-set1" description="Executes cc commands." >
  <ccsetup server="http://localhost:8090/cce"
    username="Administrator"
    password="manage"
  />
  <cc command="list landscape nodes"
    outputFormat="xml"
  />
  <cc ... />
  ...
</target>
<target name="execute-commands-set2" description="Executes cc commands.">
  <ccsetup server="http://localhost:8092/spm"
    username="Administrator"
    password="manage"
  />
  <cc command="list inventory products"
    outputFormat="json"
  />
  <cc ... />
  ...
</target>
...
```

## Parameters to Use with the `ccsetup` Task

The following table lists the parameters you can use with the `ccsetup` task to set up the base configuration for the script.

Parameter and Description
---------------------------

<code>password</code>
-----------------------

---

### Parameter and Description

Optional. Specifies the password to use for authentication on the Command Central or Platform Manager server. For example:

```
password="secret"
```

The following lists the order used to determine the value used for the password:

1. Value set with the `cc` task.
2. Value set with the `ccsetup` task.
3. Value defined in the `CC_PASSWORD` environment variable.

---

### server

Optional. Identifies the Command Central or Platform Manager server on which to execute the command. For example:

```
server="https://localhost:8092/spm"
```

The following lists the order used to determine the value used for the server:

1. Value set with the `cc` task.
2. Value set with the `ccsetup` task.
3. Value `https://localhost:8090/cce`

---

### username

Optional. Specifies the user name to use for authentication. For example:

```
username="Administrator"
```

The following lists the order used to determine the value used for the user name:

1. Value set with the `cc` task.
2. Value set with the `ccsetup` task.
3. Value defined in the `CC_USERNAME` environment variable.
4. "Administrator"

---

### trustAllHosts

Optional. Specifies whether to trust all hosts. When the parameter is included, Command Central does not verify the name of the server host. For example:

```
trustAllHosts="true"
```

The following lists the order used to determine the value for the truststore file:

1. Value set in the custom `cc.properties` file located in the `user_home \.sag` directory.
2. Value set in the Command Central default `cc.properties` file located in the `CC_CLI_HOME\conf` directory.

### Parameter and Description

#### sslTruststoreFile

Optional. Specifies the location of the truststore file. For example:

```
sslTruststoreFile=="${cce.cli.truststore.file.location}"
```

The following lists the order used to determine the value for the truststore file:

1. Value set in the custom `cc.properties` file located in the `user_home \.sag` directory.
2. Value set in the Command Central default `cc.properties` file located in the `CC_CLI_HOME\conf` directory.

#### sslTruststorePassword

Required. Specifies the password for the truststore.

```
sslTruststorePassword=="${cce.cli.truststore.password}"
```

The following lists the order used to determine the value for the truststore password:

1. Value set in the custom `cc.properties` file located in the `user_home \.sag` directory.
2. Value set in the Command Central default `cc.properties` file located in the `CC_CLI_HOME\conf` directory.

## Parameters to Use with the cc Task

The following table lists the parameters you can use with the `cc` tasks when executing commands on a Command Central server and/or a Platform Manager server.

**Note:** Beginning with Command Central and Platform Manager version 9.5.1, Software AG recommends that you use the `inputFormat` and `outputFormat` parameters in place of the `format`, `accept`, and `mediatype` parameters.

### Parameter and Description

#### accept

Deprecated. Optional. Use `outputFormat` in place of `accept`.

Specifies the format for the returned data. You supply a content type with the `accept` parameter that is used on the HTTP Accept request header sent to Command Central or Platform Manager. For example:

```
accept="json"
accept="xml"
accept="csv"
accept="tsv"
```

If you omit the `accept` parameter, `xml` is used.

## Parameter and Description

**Note:** Use either the `accept` or the `format` parameter to specify the format of the returned data. If you specify both, the value you specify with the `accept` is used.

### checkevery

Optional. Specifies the number of seconds the command waits before checking for expected output specified by the `expectedvalues` parameter. For example:

```
checkevery="10"
```

This parameter is only applicable when you also specify the `expectedvalues` parameter. If you specify the `expectedvalues` parameter but omit `checkevery`, the command uses the value of the `CC_CHECK_EVERY` environment variable. If the `CC_CHECK_EVERY` environment variable is not set, the command uses 15 seconds.

### command

Optional. Specifies a Command Central or Platform Manager command to execute. For example, to execute the following command:

```
cc list landscape nodes
```

In a script, use the following:

```
<cc command="cc list landscape nodes" />
```

Another example might be to execute the following command:

```
cc create landscape nodes alias=n1 url=localhost
```

In a script, use the following:

```
<cc command="cc create landscape nodes alias=n1
url=localhost" />
```

**Note:** Do not include the command options as described in ["Options for the Commands" on page 59](#). Instead use the corresponding attributes listed in this table. For example, if you want to specify the format "json", use `format="json"` and not `--format json`. In other words, to execute:

```
cc create landscape nodes alias=n1 url=localhost --format json
```

In a script, use the following:

```
<cc command="cc create landscape nodes alias=n1 url=localhost"
format="json"/>
```

### debug

Optional. Specifies you want extra information returned that you can use for debugging issues, in addition to the returning service output. The extra information includes:

- HTTP service request

---

### Parameter and Description

- URL of the Command Central or Platform Manager server to which the request was sent
- Request content type
- Accept header for the request
- HTTP response code from the request
- Response content type
- Response content length

---

#### **error**

Optional. Specifies the file for error output. You can specify:

- Absolute directory path and filename. For example:  
`error="c:\outputs\errors.xml"`
- Relative directory path and filename. For example:  
`error="outputs\errors.json"`
- Filename of a file in the same directory where you initiated the script. For example:  
`output="errors.xml"`

If you omit the `error` parameter, the command output is written to the console.

If you specify both the `error` and the `errorproperty` parameters, the command writes the error output to both locations identified by the parameters.

---

#### **errorproperty**

Optional. Specifies the name of a property where you want error output stored if a command fails and `failonerror="false"`. For example:

```
errorproperty="error.property"
```

If you specify both the `error` and the `errorproperty` parameters, the command writes the error output to both locations identified by the parameters.

---

#### **expectedvalues**

Optional. Specifies the expected values from a command. For example:

```
expectedvalues="STOPPED"
```

Use the `expectedvalues` parameter in conjunction with the `checkevery` and `wait` parameters.

---

## Parameter and Description

**Tip:** Using `wait="0"` with `expectedvalues` acts as a simple assertion mechanism to confirm that the output contains what you expect before executing the next step.

If you omit the `expectedvalues` parameter, the command completes without expecting a specific value.

If the expected values that you specify do not match the actual values, the build fails and stops.

---

### **failonerror**

Optional. Specifies whether to fail the entire script if an error occurs executing the command. Specify:

- `true` if you want the script to fail and stop if an error occurs.
- `false` if you want the script to continue even if the command fails. If the command fails, the error is written to the file specified with the `error` property, the `errorproperty` parameter is set with the command output, and the script can perform additional processing to check the output.

For example:

```
failonerror="false"
```

If you omit the `failonerror` parameter, command uses `true`.

---

### **format**

Deprecated. Optional. Use `outputFormat` in place of `format`.

Specifies the format you want a command to use for the data it returns. For example:

```
format="xml"
```

Command Central and Platform Manager support the following formats:

- Tab-separated values (`tsv`)
- Plain text (`txt`)
- XML (`xml`)
- Comma-separated values (`csv`)
- JavaScript Object Notation (`json`)
- ZIP (`zip`)
- PDF (`pdf`)

If you omit the `format` parameter, the command uses `xml`.

Although Command Central and Platform Manager support these formats, a specific command might only support a subset of the formats. Refer to the

### Parameter and Description

documentation for a specific command to determine the exact formats that it supports and to determine the default format for the command.

**Note:** Not all commands support plain text. If you specify `txt` for a command that does not support this format, the command uses `tsv` or `xml` based on the formats the command supports.

**Note:** Use either the `accept` or the `format` parameter to specify the format of the returned data. If you specify both, the value you specify with the `accept` is used.

### info

Optional. Sets the level of information to log to INFO.

If you omit both the `info` and `quiet` attributes, `info` is used.

### input

Required for some actions if `inputstring` is omitted. Identifies a file that contains the input data for the operation. For example, when creating a new installation, you are required to provide an alias name and URL for the installation. You would supply the alias name and URL in the input data file.

When you specify one of the following actions with the `operation` or `method` parameters, specifying `input` is required. It is not applicable for other actions.

- Operations: POST, CREATE, ADD, PUT, UPDATE, EXEC
- Methods: POST, PUT

Additionally, specifying `input` is required when using the `command` parameter if the command you specify requires input.

Supported file types for an input data file are XML (.xml), JavaScript Object Notation (.json), and properties (.properties). Although Command Central and Platform Manager support these formats, a specific command might only support a subset of the formats. Refer to the documentation for a specific command to determine the exact formats that it supports and to determine the default format for the command.

When identifying the input file, you can specify:

- Absolute directory path and filename. For example:  

```
input="c:\templates\input.xml"
```
- Relative directory path and filename. The path is relative from where you initiated the script. For example:  

```
input="templates\input.xml"
```

## Parameter and Description

- Filename of a file in the same directory where you initiated the script. For example:

```
input="input.xml"
```

### inputFormat

Optional. Specifies the content type of the input data for a command. You can specify the same values for `inputFormat` and `outputFormat`.

The default value is taken from the input file extension if the extension matches the short version of a supported media type. If the input file extension does not match the short version of a supported media type, the default is `text/plain`.

### inputstring

Required for some actions if `input` is omitted. Specifies a string that contains the actual input data for the operation.

When you specify one of the following actions with the `operation` or `method` parameters, specifying input is required. It is not applicable for other actions.

- Operations: POST, CREATE, ADD, PUT, UPDATE, EXEC
- Methods: POST, PUT

Additionally, specifying input is required when using the `command` parameter if the command you specify requires input.

For example, to change the data for the instance with ID "IS-PRIMARYPORT", for the component with ID "IntegrationServer-*instanceName*", running on the node with ID "sag01", you could use the following:

```
<cc command="update configuration data sag01
IntegrationServer-instanceName IS_PRIMARYPORT
inputstring="valid.instance.id" mediaType="text/plain"
format="txt" />
```

**Note:** Use the `inputstring` attribute when the input data is simple. For more complex data, use the `input` attribute.

### log

Optional. Specifies the file for log information. Log information is written whether commands are successful or encounter errors.

The logged results include:

- Service output
- Errors that occur while interpreting a command

**Note:** If the error occurs while the initializing a command, the error is written to the console rather than the file specified with the `log` parameter

---

## Parameter and Description

- Debug information if the `debug` parameter is specified

The log information is written to the console if you do not specify the `error` or `output` attributes.

---

### mediatype

Deprecated. Optional. Use `inputFormat` in place of `mediatype`. Specifies the content type of the input data for a command.

---

### method

Required if `operation` is omitted. Use as part of the `command` parameter. Specifies the operation to execute against a resource. For example:

```
method="PUT"
```

Command Central and Platform Manager support the following operations:

- GET to retrieve data.
- POST to add or create a new resource.
- PUT to update data for a resource.
- DELETE to delete a data.

If you omit the `method` parameter, you must specify the `operation` parameter to specify the action to execute. Use either the `method` parameter or the `operation` parameter, but not both.

---

### operation

Required if `method` is omitted. Use as part of the `command` parameter. Specifies the operation to execute against a resource. For example:

```
operation="LIST"
```

Command Central and Platform Manager support the following operations:

- GET or LIST to retrieve data.
- POST, CREATE, ADD, or EXEC to add/create a new resource or execute an action against a resource.
- PUT or UPDATE to update data for a resource.
- DELETE or REMOVE to delete data.
- OPTIONS or WADL to retrieve information for supported services.

If you omit the `operation` parameter, you must specify the `method` parameter to specify the action to execute. Use either the `method` parameter or the `operation` parameter, but not both. If you specify both, the `operation` parameter is used.

## Parameter and Description

### output

Optional. Identifies a file for command output. You can specify:

- Absolute directory path and filename. For example:  
`output="c:\outputs\results.xml"`
- Relative directory path and filename. The path is relative from where you initiated the script. For example:  
`output="outputs\results.json"`
- Filename of a file in the same directory where you initiated the script. For example:  
`output="results.xml"`

If you omit the `output` parameter, the command output is written to the console.

If you specify both the `output` and the `outputproperty` parameters, the command writes the output to both locations identified by the parameters.

### outputFormat

Optional. Specifies the format you want a command to use for the data it returns. For example:

```
outputFormat="xml"
```

Command Central and Platform Manager support the following formats:

- Tab-separated values (`tsv`)
- Plain text (`txt`)
- XML (`xml`)
- Comma-separated values (`csv`)
- JavaScript Object Notation (`json`)
- ZIP (`zip`)
- PDF (`pdf`)

The `outputFormat` parameter accepts any value for the HTTP Accept request header sent to Command Central or Platform Manager.

If you omit the `outputFormat`, but include an `-o` option in the command, Platform Manager determines the output format from the file extension. If you include a value for the `outputFormat`, for example:

```
cc list landscape nodes -p manage -output-format xml -o D:\f.json
```

Platform Manager uses the `outputFormat` value, in the example the output format will be XML.

If you omit the `outputFormat` parameter and do not include an `-o` option, the command uses `xml`.

---

### Parameter and Description

Although Command Central and Platform Manager support these formats, a specific command might only support a subset of the formats. Refer to the documentation for a specific command to determine the exact formats that it supports and to determine the default format for the command.

**Note:** Not all commands support plain text. If you specify `txt` for a command that does not support this format, the command uses `tsv` or `xml` based on the formats the command supports.

---

### **outputproperty**

Optional. Specifies an ANT property to hold the result of the command. For example:

```
outputproperty="output.property"
```

If you omit the `outputproperty`, the output is written to the console.

If you specify both the `output` and the `outputproperty` parameters, the command writes the output to both locations identified by the parameters.

---

### **password**

Optional. Specifies the password to use for authentication on the server. For example:

```
password="secret"
```

If you omit the `{--password | -p}` attribute, the command uses the value you specify with the `ccsetup` task. If you do not specify the password with the `ccsetup` task, the command uses the `CC_PASSWORD` environment variable. If the `CC_PASSWORD` environment variable is not set, the build fails indicating the password is not set.

---

### **path**

Required if `service` and `resource` are omitted. Use as part of the `command` parameter. Specifies a path that identifies the service and resource on which the command acts. To form the path, separate the service and resource by a forward slash (/) or a space. For example:

```
path="inventory/components"
```

or

```
path="inventory components"
```

**Note:** Use either the `path` parameter or the `service` and `resource` parameters to identify the service and resource on which to act.

---

### **quiet**

---

## Parameter and Description

---

Optional. Sets the level of information to log to ERROR.

---

### resource

Required if `path` is omitted. Use as part of the `command` parameter. Specifies the resource against which to execute the command. For example:

```
resource="components"
```

Examples of resources you can supply are:

- components
- environments
- fixes
- logs
- nodes
- products

When you use the `resource` parameter, you must also specify the `service` parameter to identify the service.

**Note:** Use either the `service` and `resource` parameters or the `path` parameter to identify the service and resource on which to act.

---

### responseCodeProperty

Optional. Specifies an ANT property to hold the response code. For example:

```
responseCodeProperty="response.property"
```

- If a command ends successfully, the property you specify will contain a response code that is 400 or less
  - If a command ends with an error and `failonerror` is set to `false`, the property you specify will contain an error code
- 

### server

Required if omitted from the `ccsetup` task. Identifies the server on which to execute the command. You can specify either a Command Central or Platform Manager server. For example:

```
server="http://localhost:8092/spm"
```

If you omit `server` from the `cc` task, the command uses the value you specify with the `ccsetup` task. If you omit `server` from both tasks, the command uses `http://localhost:8090/cce`.

---

### service

## Parameter and Description

Required if `path` is omitted. Use as part of the `command` parameter. Specifies the service that provides the resource on which the command acts. For example:

```
service="inventory"
```

Examples of services you can supply are:

- configuration
- diagnostics
- inventory
- jobmanager
- landscape
- lifecycle
- monitoring
- resources

When you use the `service` parameter, you must also specify the `resource` parameter to identify the resource.

**Note:** Use either the `service` and `resource` parameters or the `path` parameter to identify the service and resource on which to act.

---

### username

Optional. Specifies the user name to use for authentication. For example:

```
username="Administrator"
```

The following lists the order used to determine the value used for the user name:

1. Value set with the `cc` task.
2. Value set with the `ccsetup` task.
3. Value defined in the `CC_USERNAME` environment variable.
4. "Administrator"

---

### wait

Optional. Specifies how many seconds to wait for a long-running operation to return the expected values specified by the `expectedvalues` parameter. For example:

```
wait="160"
```

This parameter is only applicable when you also specify the `expectedvalues` parameter. If you specify the `expectedvalues` parameter but omit `wait`, the command uses the value of the `CC_WAIT` environment variable. If the `CC_WAIT` environment variable is not set, the command uses 120 seconds.

# II Working with Command Central and Platform Manager

---

■ Defining a Managed Landscape .....	97
■ Securing the Command Central Landscape .....	147
■ Managing Users, Groups, and Roles .....	169
■ Viewing Product Inventory .....	189
■ Monitoring Instances .....	221
■ Administering Product Lifecycle .....	233
■ Configuring Products .....	241
■ Repository Management .....	285
■ Template-based Provisioning .....	317
■ The Java Service Wrapper .....	355
■ Managing Command Central Licenses and Product License Reports .....	361
■ Viewing Jobs .....	383

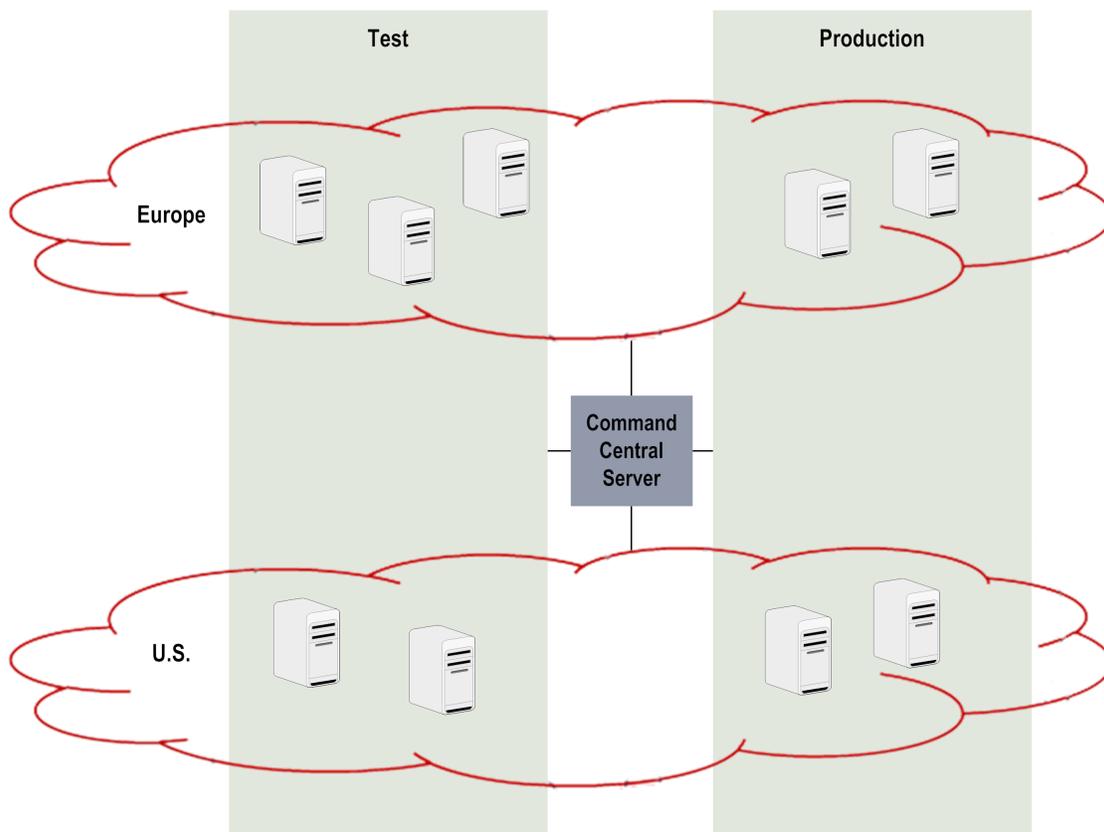


# 4 Defining a Managed Landscape

■ About Defining a Managed Landscape .....	98
■ Adding Environments .....	98
■ Filtering Environments .....	99
■ Editing Environments .....	99
■ Deleting Environments .....	100
■ Hiding/Showing Environments Pane .....	100
■ Administering Installations .....	100
■ Provisioning Bootstrap Installers Commands .....	109
■ Provisioning Products and Fixes Commands .....	117
■ Landscape Commands .....	122
■ Using Command Central to Manage Landscapes with Lower Versions .....	144

## About Defining a Managed Landscape

You manage installations by logically grouping installations under environments. This section describes how to add, view, and modify environments to define a landscape managed by Command Central. For example, the image below shows Command Central administering the installations grouped under the Test, Production, Europe, and US environments.



## Adding Environments

You add the environments that you want to centrally manage using Command Central. The default **All** environment contains the aggregate details of all the environments.

### To add a new environment to your landscape

1. In the Environments pane, click .
2. In the Add Environment dialog box, provide the following information.

In this field...	Specify...
<b>Display Name</b>	A name for the new environment. The value of this field is automatically assigned to alias. More than one environment can have the same display name, but the alias must be unique.
<b>Alias</b>	A unique name for the environment. After you create an environment, you cannot edit the alias.
<b>Description</b>	A description for the environment.

- Click **Add**. The Environments pane displays the newly added environment.

## Filtering Environments

Use filters for your Command Central landscape environments when you want to work with specific environments.

### To filter the list of environments displayed in the Environments pane

- In the **Search Environments** field, type the filter criteria.  
The Environments pane displays only the environments with display names that contain the filter text.
- Click  **Clear Filter** to clear the filter condition and display all the environments.

## Editing Environments

You can change the display name and the description of environments.

### To edit the environment details

- In the Environments pane, select the environment you want to edit and click  **Edit Environment**.
- In the Modify Environment dialog box, edit the values of the **Display Name** and **Description** fields as required.
- Click **Ok**. The changes are saved and the Environments pane displays the changes.

## Deleting Environments

---

You can remove an environment definition from your Command Central landscape. Be cautious while deleting environments because you cannot undo the environment deletion operation.

Note that when you remove an environment, the installations that were grouped under that environment are not removed from the server in which they are installed. The installations belonging to the removed environment will still be listed under the **All** environment.

To delete an environment from your Command Central landscape, in the Environments pane, select the environment you want to delete and click .

## Hiding/Showing Environments Pane

---

You can manage the view by hiding or showing the Environments pane.

To hide or show the Environments pane, click  or  respectively.

## Administering Installations

---

### About Administering Installations

You must specify the Software AG product installations you want to centrally manage through Command Central.

With Command Central, you can also create a Platform Manager installer image from a master product repository and bootstrap the Platform Manager installer to add a new Platform Manager installation on a local or remote machine. You can use the Command Central web user interface or command line interface to bootstrap Platform Manager installations.

When Command Central starts a bootstrap installation process, but a required Platform Manager bootstrap installer image does not exist, Command Central will generate a bootstrap installer image automatically during the bootstrap process, based on the installation data you specify, and register the bootstrap installer image as an image repository.

You can bootstrap a lot of Platform Manager installations at the same time. When you bootstrap more than one Platform Manager installation, you should consider the following:

- All installations on the same local or remote host are bootstrapped in sequence to ensure that only one installation session is executed at a time.
- All installations on different hosts are bootstrapped at the same time.

## Adding Installations with the Add Installation Wizard

To access the Add Installation wizard, in Command Central go to **Installations > ALL** and click . To add an installation, follow the steps, tips, and instructions in the wizard. All required fields are marked with an asterisk.

Using the Add Installation wizard, you can:

- Add an existing product installation
- Install **Platform Manager** remotely on a target machine (push installation)
- Install **Platform Manager** manually, following the instructions at the end of the Add Installation wizard (pull installation)

For all installations that you add, you must provide the **Host Name** of the host machine on which the installation is added.

You must provide the fully qualified host name or IP address of the installation, so that the products under the installation can be administered remotely. For example, `mcdev001.us.ad.gov` or `12.23.0.1`. If you want to add an installation to your local machine, you can specify `localhost`.

## Adding an Existing Installation

You add a product installation to a Command Central environment to enable Command Central to connect to and manage the products in that installation.

To add an existing installation, in the Add Installation wizard select **Platform Manager is already installed** and provide the following information:

Field	Description
<b>Port Number</b>	The port number used by the Platform Manager of the installation.
<b>Use SSL</b>	Whether the installation requires HTTP or HTTPs authentication. Select the check box to specify HTTPs.
<b>User Name</b>	A valid user name for the basic authentication credentials required by the remote Platform Manager managing the installation.

Field	Description
<b>Password</b>	A valid password for the basic authentication credentials required by the remote Platform Manager managing installation.
<b>Display Name</b>	A name for the installation. The value of this field is automatically assigned to <b>Alias</b> . More than one installation can have the same <b>Display Name</b> .
<b>Alias</b>	A unique name for the installation. No other installation in any of the environments can use this name.
<b>Description</b>	A description of the installation. This description is displayed in the Overview tab of the installation.
<b>Note:</b>	When you provide <i>HostName :PortNumber</i> of an installation that is already grouped under another environment, that installation gets linked to the environment for which you are adding. An installation with the same or different display names can be linked to more than one environment.

## Installing Platform Manager Remotely

Command Central creates a Platform Manager bootstrap installer image from which a new Platform Manager installation is installed and added remotely on a target machine.

**Note:** Ensure that the local Platform Manager is running before you start installing Platform Manager remotely.

In the Add Installation wizard, select **Install Platform Manager remotely** and the bootstrap installation method:

- **Push** - Command Central automatically installs **Platform Manager** on a remote machine.
- **Pull** - You install **Platform Manager** manually on a remote machine, following the instructions included in the **Details** step of the Add Installation wizard.

For both Push and Pull installation methods, you must provide the following information:

Field	Description
<b>Operating system</b>	The target operating system for which to create a bootstrap installer image.

Field	Description
	<p>The default value is the operating system of the machine on which Command Central is running.</p>
<b>Installation directory</b>	<p>The installation directory on the target machine in which to install the new Platform Manager installation.</p> <p>The default value is the default Software AG installation directory for the target operating system.</p>
<b>Repository</b>	<p>The master or image product repository from which to create the bootstrap installer image.</p> <p>You can use an image repository to create a bootstrap installer image when a Command Central machine does not have access to a network. The image repository is first created on a Command Central machine that can access a master repository and then registered as a product image repository on the Command Central machine that does not have network access.</p> <div data-bbox="493 947 1365 1178" style="background-color: #f0f0f0; padding: 10px;"> <p><b>Important</b> When you use a product image repository, the image repository name must follow the format:</p> <p style="text-align: center;"><i>product-distribution-version-platform</i></p> <p>Command Central automatically generates product repository names following this naming convention.</p> </div>
<b>Distribution</b>	<p>The type of software distribution to include in the bootstrap installer image.</p> <div data-bbox="493 1289 1365 1388" style="background-color: #f0f0f0; padding: 10px;"> <p><b>Tip:</b> Click the tool tip next to the <b>Distribution</b> field for a description of each distribution type.</p> </div> <ul style="list-style-type: none"> <li data-bbox="493 1409 1365 1549">■ <b>Complete</b> - Software AG recommends using the Complete distribution which contains all supported Platform Manager product plug-ins. Use Complete distribution to install and manage all supported products.</li> <li data-bbox="493 1570 1365 1669">■ <b>Default</b> - Software AG recommends this option only for experienced users. Use Default distribution when you want to select which Platform Manager product plug-ins to install.</li> <li data-bbox="493 1690 1365 1858">■ <b>Minimal</b> - Use this option only in a test or development environment. Do not use Minimal distribution in a production environment, because this distribution does not install the Java Runtime Environment (JRE) supported by webMethods products and uses the JRE of the target machine.</li> </ul>

Field	Description
<b>HTTP port</b>	<p>The HTTP port that the Platform Manager of the new installation uses.</p> <p>The default value is the default Platform Manager HTTP port.</p>
<b>HTTPs port</b>	<p>The HTTPs port that the Platform Manager of the new installation uses.</p> <p>The default value is the default Platform Manager HTTPs port.</p>
<b>Port Number</b>	<p>The port that Command Central uses to connect to the Platform Manager of the new installation.</p> <p>This field takes as default the value specified for the <b>HTTPs port</b> field.</p>
<b>User Name</b>	<p>A valid user name for the basic authentication credentials required by the remote Platform Manager managing the installation.</p> <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p><b>Note:</b> You cannot change the user name and password for the remote Platform Manager in these fields. After completing the Platform Manager bootstrap installation, you can change the basic authentication credentials in the Command Central web user interface or command line interface. For more information, see <a href="#">"Setting Up the Administrator User Password for all Platform Managers" on page 149</a>.</p> </div>
<b>Password</b>	<p>A valid password for the basic authentication credentials required by the remote Platform Manager managing the installation.</p>
<b>Display Name</b>	<p>A name for the installation. The value of this field is automatically assigned to <b>Alias</b>. More than one installation can have the same <b>Display Name</b>.</p>
<b>Alias</b>	<p>A unique name for the installation. No other installation in any of the environments can use this name.</p>
<b>Description</b>	<p>A description of the installation. This description is displayed in the Overview tab of the installation.</p>

For a Push installation only, you must also provide the following connection details to enable Command Central to connect to the target machine:

Field	Description
<b>Protocol</b>	Specifies the protocol to use to connect to the target machine. Command Central supports only the SSH protocol to connect to the remote target machine.
<b>Remote Port</b>	The remote port that Command Central uses to connect to the target machine.
<b>Authentication Method</b>	<p>The authentication method to use with the protocol.</p> <p><b>Tip:</b> Click the tool tip next to the <b>Authentication Method</b> field for a description of each authentication method.</p>
<b>User Name</b>	<p>If you select the Interactive or Password authentication methods, provide a valid user name that Command Central uses to log on the target machine.</p> <p>When Command Central connects to a remote system using SSH, ensure that the SSH user has write and execute access to the target installation directory. The SSH user should be able to manage the Platform Manager and all product instances installed in that target installation.</p> <p>To improve the security setup, Software AG recommends creating a separate SSH user account that owns the target installation directory, but otherwise has minimal rights on the target system. The installation directory should be under the user's home directory.</p>
<b>Password</b>	If you select the Interactive and Password authentication methods, provide a valid password that Command Central uses to log on the target machine.
<b>Key file</b>	If you select the Certificate authentication method, you must provide the location of the SSH key file on the Command Central server, and the user name and password required to access the key file.

After completing the Add Installation wizard:

- For a pull installation, you must follow the installation instructions provided in the wizard. Run the commands included in the instructions on the target machine and verify that Platform Manager is running and connects to Command Central successfully.

- For a push installation, Command Central automatically generates a bootstrap installer image from the master product repository or an existing image product repository. Command Central uploads the bootstrap image, installer client, and silent script on the target machine using the specified remote protocol. The installer client is executed in scripted mode with the bootstrap image.

You can monitor the bootstrap process in Command Central by clicking the **Jobs** button. Command Central will notify you when the bootstrap process is completed. You can also view the bootstrap process log from the **Logs** tab. The name of the bootstrap process log includes the job ID which you can use for troubleshooting.

When the bootstrap process is successful, the new Platform Manager installation is listed in the Command Central web user interface with  status. In the Landscape view, click  on the Installations tab to refresh the **Status** of the available installations.

### Installing Platform Manager Remotely On the Command Central Host Server

You can use the push bootstrap installation method to install Platform Managers on the Command Central host server.

---

#### To install Platform Managers on the Command Central host server

1. In the Add Installation wizard, specify `localhost` in the **Host Name** field.
2. In the Bootstrap step, select **Install Platform Manager remotely**.

The wizard automatically populates the **Operating system**, **HTTP port**, and **HTTPs port** fields with the values for the operating system and ports of the host Command Central server.

3. Specify values for the **Repository**, **Installation Directory**, and **Distribution** fields.

Note that the **Install method field** is disabled because Command Central does not require a network connection to install a Platform Manager bootstrap image on the local node.

4. Complete the Add Installation wizard steps.

Command Central adds a Platform Manager installation on the Command Central host server.

## Viewing Installations

You can view the display name, host name, port, total number of products installed in an installation node, and whether the installation is online or offline. When you expand an installation node, you can view the list of products installed in that installation.

---

#### To view the installations in an environment

1. In the Environments pane, select the environment for which you want to view the installations.

To view all the installations within all environments, select **All**.

2. Select the **Installations** tab.

The Installations tab lists the installations in the selected environment. For information about the details displayed in the Installations tab, see "[View Installations](#)" on page 35.

## Searching for Installations

Use the search filters if you want to locate specific Software AG product installations.

---

### To filter the list of installations displayed in the Installations tab

1. In the Environments pane, select the environment for which you want to filter the installations.  
To search for installations in all the environments, select **All**.
2. Select the **Installations** tab.
3. In the **Search Installations** field, type the filter text. The Installations tab displays only the installations that contain the filter text in any of its field values.

## Removing Installations

Removing an installation from an environment means that you are un-grouping that installation from the environment. If you remove an installation from all the environments (including **All**), then you cannot administer that installation through Command Central.

When you remove an installation from the **All** environment, that installation is not removed from the server; rather, it is just disconnected from Command Central.

For example, suppose that Sales installation is grouped under two environments: Testing and Production. If you remove the Sales installation from the Testing environment, you can no longer manage it from the Production and All environments; however, you can manage the Sales installation grouped under Production environment and All environment in Command Central. If you then remove the Sales installation from the Production and All environments too, you will not be able to administer the installation from anywhere in Command Central.

---

### To remove installations from an environment

1. In the Environments pane, select the environment from which you want to remove the installations.  
To view the installations of all environments, select **All**.
2. Select the **Installations** tab.

3. Select the installation you want to remove. To select multiple installations, hold down the Shift key or the Ctrl key.
4. Click  to remove the selected installations.

**Note:** Even if you remove an installation from all the defined environments, that installation is still listed in the **All** environment until you explicitly remove it from the **All** environment.

## Linking Installations to Multiple Environments

If you want an installation to be part of more than one environment, you can link that installation to multiple environments. For example, if an installation is used for both testing and development, you can link that installation to both testing and development environments.

---

### To link installations to multiple environments

1. In the Environments pane, select the environment from which you want to link the installation(s) to an environment.

To view the installations of all environments, select **All**.

2. Select the **Installations** tab.
3. Drag the installation(s) you want to link and drop it on the environment you want to link to. For selecting multiple installations, select the installations by holding down the Shift key or the Ctrl key.

You can also link an installation to multiple environments by adding the installation (same *HostName:PortNumber*) grouped under one environment to another environment. For information about adding an installation to an environment, see ["cc add landscape environments nodes" on page 122](#).

4. Click the environment to view the installations linked to it.

## Monitoring Installations

Command Central enables you to monitor the status of operating system Key Performance Indicators (KPIs) for the machine on which the installation is running.

---

### To monitor an installation

1. In the Environments pane, select the environment for which you want to monitor the installation node.
2. Select the **Installations** tab.
3. Click the name of the installation you want to monitor.
4. Select the **Overview** tab, if it is not selected.

The **Overview** tab displays the installation details, instances in the installation node, and the KPIs of the installation's Platform Manager. For more information about KPIs, see ["About Monitoring KPIs" on page 223](#).

The Installation panel displays the display name, alias name, host name, port, alias, operating system, and the version of the operating system pertaining to the installation node.

The Monitoring panel displays the KPIs of the installation's Platform Manager.

The Instances panel lists the instances of the components that are part of the installation. You can:

- View the instances of the products installed in the installation node.
- Search for instances.
- Change the status of the instances.
- Compare configurations of the instances.

## Provisioning Bootstrap Installers Commands

### cc create provisioning bootstrap installers

Command Central creates a bootstrap installer image from a local or remote master product repository, configured with valid user credentials, and registers the bootstrap installer image as an image product repository.

#### Syntax

- Command Central syntax:

- With an input file:

```
cc create provisioning bootstrap installers {--input | -i}
    filename{.xml | .json}
```

- With arguments:

```
cc create provisioning bootstrap installers
[version=version] [repoName=name]
[platform=platform] [product=product]
[distribution=type] [options]
```

- Not supported on Platform Manager.

#### Arguments and Options

Argument or Option	Description
{--input   -i} filename{.xml   .json}	Optional. Identifies an input file in XML or JSON format that contains the bootstrap installer image

Argument or Option	Description
	<p>details. For more information, see "<a href="#">--input   -i</a>" on <a href="#">page 70</a>.</p> <p>You can either use an input data file or specify arguments for the command. For an example of a bootstrap details XML file, see the usage notes for the <a href="#">cc exec provisioning bootstrap nodes</a> command.</p>
<p>[<i>version=version</i>]</p>	<p>Optional. The version of the product repository for which to create the bootstrap installer image.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> <li>■ <i>major</i> version, for example 9</li> <li>■ <i>major.minor</i> version, for example 9.6</li> <li>■ <i>latest</i></li> </ul> <p>If Command Central cannot match the specified version to a registered product repository, Command Central returns an error.</p> <p>When you do not include this parameter, the command uses the latest version available in the product repository.</p>
<p>[<i>repoName=name</i>]</p>	<p>Optional. The name of the product repository from which to create the bootstrap installer image.</p> <p>When you do not include this parameter, the command uses:</p> <ul style="list-style-type: none"> <li>■ The product repository that matches the version specified in the <i>version=version</i> argument</li> <li>■ If the <i>version=version</i> argument is not specified, Command Central uses the registered repository for the latest version.</li> <li>■ If only one product repository is registered in Command Central, the command uses that repository.</li> <li>■ If no repositories are registered in Command Central, the command returns an error.</li> </ul>
<p>[<i>platform=platform</i>]</p>	<p>Optional. The ID of the target operating system for which to create the bootstrap installer image.</p>

Argument or Option	Description
	When you do not include this parameter, the command uses the operating system on the localhost machine.
[ <i>product=product</i> ]	<p>Optional. The ID of products to include in the bootstrap installer image. Valid values are:</p> <ul style="list-style-type: none"> <li>■ SPM (default)</li> <li>■ CCE</li> <li>■ CLI</li> </ul>
[ <i>distribution=type</i> ]	<p>The type of software distribution to include in the bootstrap installer image. Valid values are:</p> <ul style="list-style-type: none"> <li>■ MIN. Platform Manager only.</li> <li>■ DEF. Platform Manager, Update Manager and the Command Central command line interface.</li> <li>■ ALL (default). Platform Manager, Update Manager, the Command Central command line interface, and all Platform Manager plug-ins.</li> </ul> <p>If you do not specify a value, the command includes the complete distribution.</p>
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Examples When Executing on Command Central

- To create a bootstrap installer image with complete distribution from the latest product version for the operating system on the localhost machine:
 

```
cc create provisioning bootstrap installers
```
- To create a bootstrap installer image with complete distribution from a repository with version "9.6" for operating system with ID "LNXADM64":
 

```
cc create provisioning bootstrap installers
version=9.6 platform=LNXADM64 distribution=ALL
```
- To create a bootstrap installer image, based on the "bootstrapInfo.xml" file that contains the bootstrap data required to create the image:
 

```
cc create provisioning bootstrap installers --input bootstrapInfo.xml
```

## Related Commands

[cc list provisioning bootstrap installers](#)

[cc exec provisioning bootstrap nodes](#)

## cc list provisioning bootstrap installers

Retrieves a list of available bootstrap installer images.

### Syntax

- Command Central syntax:

```
cc list provisioning bootstrap installers [installer Name]
  [version=version] [platform=platform] [product=product]
  [distribution=type] [options]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>installerName</i>	Optional. The name of an existing bootstrap installer image for which you want to search. For example, <i>spm-all-9.6-lnxadm64.zip</i>
[ <i>version=version</i> ]	Optional. The version of the bootstrap installer image. Valid values are: <ul style="list-style-type: none"> <li>■ <i>major</i> version, for example 9</li> <li>■ <i>major.minor</i> version, for example 9.6</li> <li>■ <i>latest</i></li> </ul>
[ <i>platform=platform</i> ]	Optional. The ID of the target operating system of the bootstrap installer image.
[ <i>product=product</i> ]	Optional. The ID of products to include in the bootstrap installer image. Valid values are: <ul style="list-style-type: none"> <li>■ SPM</li> <li>■ CCE</li> <li>■ CLI</li> </ul>

Argument or Option	Description
[distribution= <i>type</i> ]	The type of software distribution included in the bootstrap installer image. Valid values are: <ul style="list-style-type: none"> <li>■ MIN. Platform Manager only.</li> <li>■ DEF. Platform Manager, Update Manager and the Command Central command line interface.</li> <li>■ ALL. Platform Manager, Update Manager, the Command Central command line interface, and all Platform Manager plug-ins.</li> </ul>
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Examples When Executing on Command Central

- To list all available bootstrap installer images:

```
cc list provisioning bootstrap installers
```
- To list the existing bootstrap installer images with “Minimal” distribution for repository version “9.6” and operating system “OSX”:

```
cc list provisioning bootstrap installers
version=9.6 platform=OSX distribution=MIN
```

### Related Commands

[cc create provisioning bootstrap installers](#)

[cc exec provisioning bootstrap nodes](#)

## cc exec provisioning bootstrap nodes

Installs a bootstrap installer image on a local or remote machine and registers the new Platform Manager installation under a specified node alias.

### Syntax

- Command Central syntax:
  - With an input file:

```
cc exec provisioning bootstrap nodes nodeAlias [--input | -i]
filename{.xml | .json}
```
  - With arguments:

```
cc exec provisioning bootstrap nodes nodeAlias [hostname=hostname]
[repoName=name] [version=version] [distribution=type]
```

```
[platform=platform] [installDir=directory] [httpPort=port]
[httpsPort=port] [options]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<pre>{--input   -i} filename{.xml   .json}</pre>	<p>Optional. Identifies an input file in XML or JSON format that contains the bootstrap installation details. For more information, see "<a href="#">--input   -i</a>" on <a href="#">page 70</a>.</p> <p>You can either use an input data file or specify arguments for the command. For an example of a bootstrap details XML file, see the Usage Notes section.</p>
<pre>nodeAlias</pre>	<p>Required. Specifies the alias name of the bootstrapped Platform Manager installation that you want to add to the target machine.</p>
<pre>[hostname=<i>hostname</i>]</pre>	<p>Optional. The host name of the target machine on which to install the bootstrap installer image. The default is <code>localhost</code>.</p>
<pre>[repoName=<i>name</i>]</pre>	<p>Optional. The name of the product repository from which the bootstrap installer image is created.</p> <p>When you do not include this parameter, the command uses:</p> <ul style="list-style-type: none"> <li>■ The product repository that matches the version specified in the <code>version=<i>version</i></code> argument</li> <li>■ If the <code>version=<i>version</i></code> argument is not specified, Command Central uses the registered repository for the latest version.</li> <li>■ If only one product repository is registered in Command Central, the command uses that repository.</li> <li>■ If no repositories are registered in Command Central, the command returns an error.</li> </ul>

Argument or Option	Description
[ <i>version=version</i> ]	<p>Optional. The version of the product repository from which the bootstrap installer image is created.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> <li>■ <i>major</i> version, for example 9</li> <li>■ <i>major.minor</i> version, for example 9.6</li> <li>■ <i>latest</i></li> </ul> <p>If Command Central cannot match the specified version to a registered product repository, Command Central returns an error.</p> <p>When you do not include this parameter, the command uses the latest version available in the product repository.</p>
[ <i>distribution=type</i> ]	<p>Optional. The type of software distribution included in the bootstrap installer image. Valid values are:</p> <ul style="list-style-type: none"> <li>■ MIN. Platform Manager only.</li> <li>■ DEF. Platform Manager, Update Manager and the Command Central command line interface.</li> <li>■ ALL. Platform Manager, Update Manager, the Command Central command line interface, and all Platform Manager plug-ins.</li> </ul> <p>If you do not specify a value, the command includes the complete distribution.</p>
[ <i>platform=platform</i> ]	<p>Optional. The ID of the target operating system of the bootstrap installer image.</p> <p>When you do not include this parameter, the command uses the operating system on the localhost machine.</p>
[ <i>installDir=directory</i> ]	<p>Optional. The directory in which to install the bootstrap Platform Manager installation.</p> <p>The default values are platform specific:</p> <ul style="list-style-type: none"> <li>■ Windows: c:\SoftwareAG</li> <li>■ UNIX: /opt/softwareag</li> </ul>

Argument or Option	Description
	<ul style="list-style-type: none"> <li>■ OSX: /Applications/SoftwareAG</li> </ul> <p><b>Note:</b> When specifying a value for a Windows directory, enclose the value in double quotes.</p>
[httpPort= <i>por</i> ]	Optional. The HTTP port for the bootstrap Platform Manager installation. The default is 8092.
[httpsPort= <i>port</i> ]	Optional. The HTTPS port for the bootstrap Platform Manager installation. The default is 8093.
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

- After you execute the `cc exec provisioning bootstrap nodes` command, use the `cc get landscape nodes` command to verify the node status. For example:

```
cc get landscape nodes sag01 --expected-values ONLINE
--wait 60 --check-every 5
```

where "sag01" is the alias name of the bootstrap Platform Manager installation.

When the execution of the `cc get landscape nodes` command is successful, you can apply templates to the bootstrap Platform Manager installation. For more information about the `--expected-values`, `--wait`, and `--check-every` options see ["--expected-values | -e" on page 64](#), ["--wait | -w" on page 80](#), and ["--check-every | -c" on page 60](#).

- Following is an example of a bootstrap details XML file required for installing a bootstrap Platform Manager installation:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<bootstrapInfo>
  <repoName>webMethods-9.6_US</repoName>
  <version>latest.integration</version>
  <platform>OSX</platform>
  <distribution>ALL</distribution>
  <useImage>true</useImage>
  <hostname>localhost</hostname>
  <protocol>SSH</protocol>
  <port>22</port>
  <credentials>
    <authenticationType>BASIC</authenticationType>
    <authenticationMethod>INTERACTIVE</authenticationMethod>
    <userName>Administrator</userName>
  </credentials>
</bootstrapInfo>
```

```

    <password>manage</password>
  </credentials>
  <installDir>/some/dir</installDir>
  <installJava>>false</installJava>
  <httpPort>1234</httpPort>
  <httpsPort>1235</httpsPort>
</bootstrapInfo>

```

Valid values for the `<authenticationMethod>` parameter are `PASSWORD` (default), `INTERACTIVE`, and `CERTIFICATE`.

### Examples When Executing on Command Central

- To install a Platform Manager from the latest version of a bootstrap image generated from a node with alias name “sag01”, HTTP port “8092”, and HTTPS port “8093”, in the “c:\install\test” directory on a local node:

```

cc exec provisioning bootstrap nodes sag01
installDir=c:\install\test httpPort=8092 httpsPort=8093

```

- To install a Platform Manager from a bootstrap image with alias name “sag01”, default authentication credentials, and default installation details (such as ports, installation directory, and distribution type) on a remote node with host name “rubicon” and operating system “LNXADM64”:

```

cc exec provisioning bootstrap nodes sag01
hostname=rubicon platform=LNXADM64

```

- Bootstrap installation details are in the `bootstrapInfo.xml` file. To install a Platform Manager from a bootstrap image with alias name “sag01”, using the details in the `bootstrapInfo.xml` file:

```

cc exec provisioning bootstrap nodes sag01 --input bootstrapInfo.xml

```

### Related Commands

[cc create provisioning bootstrap installers](#)

[cc list provisioning bootstrap installers](#)

[cc get landscape nodes](#)

## Provisioning Products and Fixes Commands

### cc exec provisioning fixes install

Installs fixes from a fix repository. The Platform Manager on which you install the fixes must have access to Empower or an image fix repository.

#### Syntax

- Command Central syntax:

```

cc exec provisioning fixes node_alias repo name install
[products=productId,productId2]
[artifacts=fixName1[_version],fixName2[_version]] [options]

```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<code>node_alias</code>	Required. The alias of the target node on which to install fixes.
<code>repo name</code>	Required. The name of the fix repository from which to install fixes.
<code>[products= productId, productId2]</code>	Optional. The IDs of the products for which you want to install fixes.
<code>[artifacts= fixName1 [_version], fixName2 [_version]]</code>	Optional. The names of the fixes to install. Specifying the fix version is optional.
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Examples When Executing on Command Central

- To install all fixes in a master fix repository with name "repo1" for all products installed on a target node with alias "sag01":
 

```
cc exec provisioning fixes sag01 repo1 install
```
- To install all fixes from an image repository with name "MyFixes" for all products installed on a target node with alias "sag01":
 

```
cc exec provisioning fixes sag01 MyFixes install
```
- To install the fixes with names "wMFix.SPM.TEST1" and "wMFix.SPM.TEST2" and their dependencies from a master repository with name "repo1" on a remote node with alias "sag01":
 

```
cc exec provisioning fixes sag01 repo1 install  
artifacts=wMFix.SPM.TEST1,wMFix.SPM.TEST2
```
- To install all fixes in a fix repository with name "repo1" for the products with IDs "SPM" and "OSGI":
 

```
cc exec provisioning fixes sag01 repo1 install  
products=SPM,OSGI
```

## Related Commands

[cc exec provisioning products install](#)

[cc exec provisioning uninstall](#)

## cc exec provisioning products install

Installs products from a product repository. The Platform Manager on which you install the products must have access to a master or image product repository.

### Syntax

- Command Central syntax:

```
cc exec provisioning products nodeAlias repoName install
[artifacts=productId[_version],productId[_version]] [options]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	Required. The alias of the target node on which to install products.
<i>repo_name</i>	Required. The name of the product repository from which to install products.
[artifacts= <i>productId</i> [ <i>_version</i> ], <i>productId</i> [ <i>_version</i> ]	<p>The names of the products to install.</p> <ul style="list-style-type: none"> <li>■ Required for a master repository. You must specify at least one product with version.</li> <li>■ Optional for image repositories. If you do not specify any artifacts, Command Central installs all products in the image archive.</li> </ul> <p>When you specify more than one product, specifying the version is optional.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p><b>Note:</b> To find the product IDs, use the <a href="#">cc list repository products content</a> command.</p> </div>
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line

Argument or Option	Description
	Interface. For a description of the options, see <a href="#">"Options for the Commands"</a> on page 59.

### Usage Note

You cannot install all products available in a master repository. You must specify at least one product with version in the `artifacts` argument.

### Examples When Executing on Command Central

- To install all products from an image repository with name “MyProducts” on a target node with alias “sag01”:

```
cc exec provisioning products sag01 MyProducts install
```

- To install the products with product IDs “integrationServer” and “MwsProgramFiles” and their dependencies from a master repository with name “webMethods-EUR” on a remote node with alias “sag01”:

```
cc exec provisioning products sag01 webMethods-EUR install
artifacts=integrationServer_9.8,MwsProgramFiles
```

### Related Commands

[cc exec provisioning fixes install](#)

[cc exec provisioning uninstall](#)

[cc list repository products content](#)

## cc exec provisioning uninstall

Uninstall a product or a fix. The uninstall operation does not require access to a repository.

**Note:** A product depends on its Platform Manager plug-in. You cannot uninstall a product without uninstalling the Platform Manager plug-in for that product.

### Syntax

- Command Central syntax:

```
cc exec provisioning {products | fixes}
    node_alias uninstall [artifacts=Id1,Id2]
    [options]
```

- Not supported on Platform Manager.

## Arguments and Options

Argument or Option	Description
<code>node_alias</code>	Required. The alias of the installation from which to uninstall products or fixes.
<code>[artifacts=Id1, Id2]</code>	Optional. The IDs of the products or fixes that you want to uninstall. If you omit the artifacts argument, Command Central uninstalls all products, including Platform Manager.
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

## Usage Notes

- When uninstalling a fix, Platform Manager restores the previously installed version of the fix. For example, when you uninstall SPM\_9.7\_Fix2, the command rolls back SPM\_9.7\_Fix1.
- If a fix depends on other fixes, uninstalling the fix fails with an error message that lists the fix dependencies. You must include all listed fix dependencies in the `artifacts` argument and run the command again.

## Examples When Executing on Command Central

- To uninstall a fix with ID "wMFix.SPM.TEST" from a node with alias "sag01":  

```
cc exec provisioning fixes sag01 uninstall artifacts=wMFix.SPM.TEST
```
- To uninstall all fixes from a node with alias "sag01":  

```
cc exec provisioning fixes sag01 uninstall
```
- To uninstall a product with ID "integrationServer" from a node with alias "sag01":  

```
cc exec provisioning products sag01 uninstall artifacts=integrationServer,ISspm
```
- To uninstall all products, including Platform Manager, from a node with alias "sag01":  

```
cc exec provisioning products sag01 uninstall
```

## Related Commands

[cc exec provisioning fixes install](#)

[cc exec provisioning products install](#)

## Landscape Commands

### cc add landscape environments nodes

Adds one or more existing installations (also known as *nodes*) to a specified environment.

#### Syntax

##### ■ Command Central syntax:

```
cc add landscape environments env_alias
nodes nodeAlias=alias1 [nodeAlias=alias2 ... nodeAlias=aliasn]
[options]
```

```
options:
[--debug | -d]
[--error | -r] file
[--log | -l] file
[--password | -p] password
[--quiet | -q]
[--server | -s] url]
[--username | -u] user_name]
```

##### ■ Not applicable to Platform Manager

#### Arguments and Options

Argument or Option	Description
<i>env_alias</i>	Required. Specifies the alias name of the environment to which you want to add one or more installations.
<i>nodes</i>	Required. Specifies a required keyword indicating you are adding installations (also known as <i>nodes</i> ) to an environment.
<i>nodeAlias=alias1</i> [ <i>nodeAlias=alias2</i> ... <i>nodeAlias=aliasn</i> ]	Required. Specifies the alias name(s) of one or more installation(s) that you want to add to the environment.
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

## Usage Notes

- Use the [cc create landscape environments](#) command to create an environment.
- Use the [cc create landscape nodes](#) command to create installations that Command Central manages and that you can then add to an environment.
- If you specify installation alias names both on the command line and in an input data file using the `{--input | -i}` option, the command ignores the alias names on the command line and uses only those specified in the input data file.
- You can add the same installation to multiple environments.

## Examples When Executing on Command Central

In the following commands the `{--server | -s}`, `{--username | -u}`, and `{--password | -p}` options are not specified. As a result, the command uses the default server, user name, and password. For more information, see "[--server | -s](#)" on page 77, "[--username | -u](#)" on page 80, and "[--password | -p](#)" on page 76.

- To add the installation with alias name "sag01" to the environment with alias name "dev1":

```
cc add landscape environments dev1 nodes nodeAlias=sag01
```

- To add the installations with alias names "is02" and "mws02" to the environment with alias name "env2":

```
cc add landscape environments env2 nodes nodeAlias=is02 nodeAlias=mws02
```

## Related Commands

[cc create landscape environments](#)

[cc get landscape environments](#)

[cc list landscape environments](#)

[cc remove landscape environments nodes](#)

[cc create landscape nodes](#)

[cc get landscape nodes](#)

[cc list landscape nodes](#)

## cc create landscape environments

Creates a new environment that you want to use to manage a collection of installations.

### Syntax

- Command Central syntax:
  - To specify the data for the new environment on the command line:

```
cc create landscape environments alias=env_alias [name=name]
```

```
[description=description] [options]
```

- To specify the data for the new environment in an input data file:

```
cc create landscape environments
  --input | -i} filename{.xml|.json} [options]
```

```
options:
[--debug | -d]
[--error | -r} file]
[--log | -l} file]
[--media-type | -m} content-type]
[--password | -p} password]
[--quiet | -q]
[--server | -s} url]
[--username | -u} user_name]
```

- Not applicable to Platform Manager.

## Arguments and Options

Argument or Option	Description
<code>alias=<i>env_alias</i></code>	<p>Required. Specifies the alias name you want to assign to the new environment. The name must be unique among all environments that Command Central manages.</p> <p>Valid characters are ASCII characters, numbers, hyphen (-), underscore (_), and period (.). Spaces are not allowed.</p>
<code>[name=<i>name</i>]</code>	<p>Optional. Specifies the display name you want to assign to the environment. If you use a value that includes spaces, place quotes around the value, for example:</p> <pre>name="Dev Environment"</pre> <p>If you do not specify a display name, the command uses the value you supply for the environment alias name.</p>
<code>[description=<i>description</i>]</code>	<p>Optional. Specifies a description for the new environment. If you use a value that includes spaces, place quotes around the value, for example:</p> <pre>description="A description with spaces"</pre>
<code>[<i>--input</i>   -i} <i>filename</i>{.xml .json}]</code>	<p>Optional. Identifies an input file that contains the data for the new environment. For more information, see "<a href="#">--input   -i</a>" on page 70.</p>

Argument or Option	Description
	<p><b>Tip:</b> To determine how to specify the data in the input file, use <a href="#">cc get landscape environments</a> to retrieve data for an existing environment. For example, if you want to use an XML input file, use <a href="#">cc get landscape environments</a> with the <code>--format xml</code> option to retrieve the data in XML format.</p>
<code>[options]</code>	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

- After creating the new environment, use [cc add landscape environments nodes](#) to add existing installations to the environment.

### Example When Executing on Command Central

To create a new environment with the display name “Development1”, the alias name “dev1”, and a description “Environment to test latest release”:

```
cc create landscape environments name=Development1 alias=dev1
description="Environment to test latest release" --password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see ["--server | -s" on page 77](#) and ["--username | -u" on page 80](#). The command specifies “secret” for the user’s password

### Related Commands

[cc add landscape environments nodes](#)

[cc delete landscape environments](#)

[cc get landscape environments](#)

[cc list landscape environments](#)

[cc remove landscape environments nodes](#)

[cc update landscape environments](#)

## cc delete landscape environments

Deletes a specified environment.

## Syntax

### ■ Command Central syntax:

```
cc delete landscape environments [env_alias] [options]

options:
[--debug | -d]
[--error | -r] file
[--force]
[--log | -l] file
[--password | -p] password
[--quiet | -q]
[--server | -s] url
[--username | -u] user_name
```

### ■ Not applicable to Platform Manager.

## Arguments and Options

Argument or Option	Description
<i>env_alias</i>	Optional. Specifies the alias name of the environment you want to delete.
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

## Usage Notes

- If you omit *env\_alias*, the command deletes all environments.
- When you delete an environment, the installations in that environment are not deleted. They are still under Command Central management, but no longer assigned to the environment.
- If you want to remove an installation from the environment, use the [cc remove landscape environments nodes](#) command.
- If you want to remove an installation from Command Central management, use the [cc delete landscape nodes](#) command.

## Example When Executing on Command Central

To delete the environment with the alias name “dev1” using the authorization of the user with user name “Administrator” and password “manage”:

```
cc delete landscape environments dev1 --username Administrator
--password manage
```

Because the `{--server | -s}` option is not specified, the command uses the default server. For more information, see ["--server | -s" on page 77](#).

## Related Commands

[cc create landscape environments](#)

[cc get landscape environments](#)

[cc list landscape environments](#)

[cc remove landscape environments nodes](#)

[cc update landscape environments](#)

[cc delete landscape nodes](#)

## cc get landscape environments

Retrieves information about a specified environment. Information about an environment can include:

- Alias name
- Display name
- Description, or null if none is assigned
- Information about the installations in the environment

### Syntax

- Command Central syntax:

```
cc get landscape environments env_alias [nodes] [options]
```

```
options:
[--accept | -a] content_type
[--debug | -d]
[--error | -r] file
[--format | -f] {tsv args | text | xml | csv args | json}
[--log | -l] file
[--output | -o] file
[--password | -p] password
[--quiet | -q]
[--server | -s] url
[--username | -u] user_name
```

- Not applicable to Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>env_alias</i>	Required. Specifies the alias name of the environment whose information you want to retrieve.

Argument or Option	Description
[nodes]	<p>Optional. Indicates you want the command to return the information about the installations in the environment.</p> <p>If you omit the <code>nodes</code> parameter, the returned information will not include the list of installations in the environment.</p>
[options]	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a>.</p>

### Examples When Executing on Command Central

- To execute a command on the Command Central server with host name “rubicon” and port “8090” to retrieve information for the “dev1” environment, using the authorization of the user with user name “Administrator” and password “manage”, and have the information returned to the console in XML format:

```
cc get landscape environments dev1 --format xml --server
http://rubicon:8090/cce --username Administrator --password manage
```

- To execute a command on the Command Central server with host name “rubicon” and port “8090” to retrieve information for the “dev1” environment and include information about its installations, using the authorization of the user with user name “Administrator” and password “manage”, and have the information returned to the console in JavaScript Object Notation format:

```
cc get landscape environments dev1 --format json --server
http://rubicon:8090/cce --username Administrator --password manage
```

### Related Commands

[cc create landscape environments](#)

[cc delete landscape environments](#)

[cc list landscape environments](#)

[cc update landscape environments](#)

## cc list landscape environments

Lists environments in the landscape. Information about an environment can include:

- Alias name
- Display name if one is assigned; otherwise null
- Description if one is assigned; otherwise null
- List of installation aliases that belong to the environment

## Syntax

### ■ Command Central syntax:

```
cc list landscape environments [env_alias] [options]

options:
[--accept | -a] content_type
[--debug | -d]
[--error | -r] file
[--format | -f] {tsv args | text | xml | csv args | json}
[--log | -l] file
[--output | -o] file
[--password | -p] password
[--quiet | -q]
[--server | -s] url
[--username | -u] user_name
```

### ■ Not applicable to Platform Manager.

## Arguments and Options

Argument or Option	Description
[ <i>env_alias</i> ]	Optional. Specifies the alias name of the environment whose information you want to retrieve. If you do not specify an alias name, the command lists information for all environments.
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

## Example When Executing on Command Central

To list all environments that the Command Central with host name “rubicon” and port “8090” manages, using the authorization of the user with user name “Administrator” and password “manage”, and have the information returned to the output file “environlist” in XML format:

```
cc list landscape environments --format xml --output environlist
--server http://rubicon:8090/cce --username Administrator --password manage
```

## Related Commands

[cc create landscape environments](#)

[cc delete landscape environments](#)

[cc get landscape environments](#)

## cc update landscape environments

## cc remove landscape environments nodes

Removes one or more installations from a specified environment.

### Syntax

- Command Central syntax:

```
cc remove landscape environments env_alias
[nodes nodeAlias=alias1 [nodeAlias=alias2 ... nodeAlias=aliasn]] [options]

  options:
  [--debug | -d]
  [--error | -r} file]
  [--log | -l} file]
  [--password | -p} password]
  [--quiet | -q]
  [--server | -s} url]
  [--username | -u} user_name]
```

- Not applicable to Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>env_alias</i>	Required. Specifies the alias name of the environment from which you want to remove one or more installations.
nodeAlias= <i>alias1</i> [nodeAlias= <i>alias2</i> ... nodeAlias= <i>aliasn</i> ]	Optional. Specifies the alias name(s) of one or more installations that you want to remove from the environment.  If you do not specify alias names, the command removes all installations from the specified environment.
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

- The installations that you remove from the environment are not deleted or uninstalled. They are still managed by Command Central, but are no longer associated with the environment.

- If you want to remove an installation from Command Central management, use the `cc delete landscape nodes` command.

### Example When Executing on Command Central

To remove the installations with alias names “mws02” and “is02” from the environment with alias name “env2”:

```
cc remove landscape environments env2 nodeAlias=mws02 nodeAlias=is02
--password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see “[--server | -s](#)” on page 77 and “[--username | -u](#)” on page 80. The command specifies “secret” for the user’s password

### Related Commands

[cc add landscape environments nodes](#)

[cc create landscape environments](#)

[cc delete landscape environments](#)

[cc get landscape environments](#)

[cc list landscape environments](#)

[cc update landscape environments](#)

[cc delete landscape nodes](#)

[cc get landscape nodes](#)

[cc list landscape nodes](#)

## cc update landscape environments

Updates the display name and/or description assigned to an existing environment.

### Syntax

- Command Central syntax:

- To specify the updated data for the environment on the command line:

```
cc update landscape environments env_alias [name=name]
[description=description] [options]
```

- To specify the updated data for the environment in an input data file:

```
cc update landscape environments env_alias
--input | -i filename{.xml|.json} [options]
```

```
options:
[--debug | -d]
[--error | -r] file
[--log | -l] file
```

```
[--media-type | -m] content-type ]
[--password | -p] password ]
[--quiet | -q] ]
[--server | -s] url ]
[--username | -u] user_name ]
```

- Not applicable to Platform Manager.

## Arguments and Options

Argument or Option	Description
<i>env_alias</i>	Required. Specifies the alias name of the environment whose description you want to update.
[ <i>name=name</i> ]	Optional. Specifies the updated display name for the environment.  If you use a value that includes spaces, place quotes around the value, for example: <code>name="Dev Environment"</code>
[ <i>description=description</i> ]	Optional. Specifies the updated description for the environment.  If you use a value that includes spaces, place quotes around the value, for example: <code>description="A description with spaces"</code>
[ <i>--input   -i</i> ] <i>filename</i> {.xml .json}]	Optional. Identifies an input file that contains the updated data for the environment. For more information, see " <a href="#">--input   -i</a> " on page 70.  <b>Tip:</b> To determine how to specify the data in the input file, use <a href="#">cc get landscape environments</a> to retrieve data for the environment you want to update. For example, if you want to use an XML input file, use <a href="#">cc get landscape environments</a> with the <code>--format xml</code> option to retrieve the data in XML format.
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

## Usage Notes

- You must specify at least one of the `name` or `description` arguments to indicate the item that you want to update for the environment.

## Example When Executing on Command Central

To update the description of an environment with the alias name “dev1” to use the description, “Development version”, use the following command:

```
cc update landscape environments dev1 description="Development version"
--password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see “[--server | -s](#)” on page 77 and “[--username | -u](#)” on page 80. The command specifies “secret” for the user’s password.

## Related Commands

[cc create landscape environments](#)

[cc delete landscape environments](#)

[cc get landscape environments](#)

[cc list landscape environments](#)

## cc create landscape nodes

Adds an installation (also known as a *node*) that you want to manage via Command Central.

### Syntax

- Command Central syntax:

- To specify the data for the new landscape on the command line:

```
cc create landscape nodes alias=node_alias url=url [name=name]
[description=description] [options]
```

- To specify the data for the new landscape in an input data file:

```
cc create landscape nodes [--input | -i] filename{.xml|.json}
[options]
```

```
options:
[{--debug | -d}]
[{--error | -r} file]
[{--log | -l} file]
[{--media-type | -m} content-type]
[{--password | -p} password]
[{--quiet | -q}]
[{--server | -s} url]
[{--username | -u} user_name]
```

- Not applicable to Platform Manager.

## Arguments and Options

Argument or Option	Description
<code>alias=node_alias</code>	<p>Required. Specifies the alias name you want to assign to the installation. The name must be unique among all installations that Command Central manages.</p> <p>Valid characters are ASCII characters, numbers, hyphen (-), underscore (_), and period (.). Spaces are not allowed.</p>
<code>url=url</code>	<p>Required. Specifies the URL of the Command Central that manages the installation. For example:</p> <pre>http://rubicon:8092</pre> <p>When specifying the URL, if you omit the port number, the command uses "8092", which is the default port for a Command Central server.</p> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><b>Note:</b> Do not specify a URL that uses the HTTPS protocol. The HTTPS protocol is not supported.</p> </div>
<code>[name=name]</code>	<p>Optional. Specifies the display name you want to assign to the installation. If you use a value that includes spaces, place quotes around the value, for example:</p> <pre>name="my installation"</pre> <p>If you do not specify a display name, the command uses the value you specify for the alias name.</p>
<code>[description=description]</code>	<p>Optional. Specifies a description for the installation. If you use a value that includes spaces, place quotes around the value, for example:</p> <pre>description="A description with spaces"</pre>
<code>[{--input   -i} filename{.xml .json}]</code>	<p>Optional. Identifies an input file that contains the data for the new node. For more information, see "<a href="#">--input   -i</a>" on page 70.</p>

Argument or Option	Description
	<p><b>Tip:</b> To determine how to specify the data in the input file, use <code>cc get landscape nodes</code> to retrieve data for an existing node. For example, if you want to use an XML input file, use <code>cc get landscape nodes</code> with the <code>--format xml</code> option to retrieve the data in XML format.</p>
[options]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Usage Notes

- Use the `cc create landscape nodes` command to create an installation that is not associated with an environment. After creating the installation, you can use the `cc add landscape environments nodes` command to associate the installation with an environment.

### Example When Executing on Command Central

To add an installation managed by the Platform Manager with the URL “http://spm:8092”, and assign it the display name “My webMethods Server” and alias name “mws01”:

```
cc create landscape nodes name="My webMethods Server" alias=mws01
url=http://spm:8092 --password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see "[--server | -s](#)" on page 77 and "[--username | -u](#)" on page 80. The command specifies “secret” for the user’s password

### Related Commands

[cc delete landscape nodes](#)

[cc get landscape nodes](#)

[cc list landscape nodes](#)

[cc update landscape nodes](#)

[cc add landscape environments nodes](#)

## cc delete landscape nodes

Removes an installation from being centrally managed via Command Central.

## Syntax

### ■ Command Central syntax:

```
cc delete landscape nodes [{alias | nodeAlias=alias1
[nodeAlias=alias2 ... nodeAlias=aliasn]}] [options]

options:
[--debug | -d]
[--error | -r] file
[--force]
[--log | -l] file
[--password | -p] password
[--quiet | -q]
[--server | -s] url
[--username | -u] user_name
```

### ■ Not applicable to Platform Manager.

## Arguments and Options

Argument or Option	Description
<pre>[{alias   nodeAlias=alias1 [nodeAlias=alias2 ... nodeAlias=aliasn]}</pre>	<p>Optional. Specifies the alias name(s) of the installation(s) you want to remove.</p> <p>If you execute the <code>cc delete landscape nodes</code> command without supplying alias names, the command removes all installations it is currently managing.</p> <p>To remove a single installation, supply its alias name using the following format:</p> <pre>cc delete landscape nodes alias</pre> <p>To remove multiple installations, supply each name using the <code>nodeAlias=alias</code> format to identify each node to remove.</p>
<pre>[options]</pre>	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a>.</p>

## Usage Notes

- The `cc delete landscape nodes` command does not physically delete the installation(s). It just removes the installation(s) from Command Central management.

- To remove an installation from a specific environment, use the [cc remove landscape environments nodes](#) command.

### Examples When Executing on Command Central

In the following commands the `{--server | -s}`, `{--username | -u}`, and `{--password | -p}` options are not specified. As a result, the command uses the default server, user name, and password. For more information, see "[--server | -s](#)" on page 77, "[--username | -u](#)" on page 80, and "[--password | -p](#)" on page 76.

- To remove the installation with alias "mws01":
 

```
cc delete landscape nodes mws01
```
- To remove the installations with alias names "mws01" and "sag01":
 

```
cc delete landscape nodes nodeAlias=mws01 nodeAlias=sag01
```
- To remove all installations:
 

```
cc delete landscape nodes
```

### Related Commands

[cc create landscape nodes](#)

[cc get landscape nodes](#)

[cc list landscape nodes](#)

[cc update landscape nodes](#)

[cc remove landscape environments nodes](#)

## cc exec landscape nodes generateNodeId

Generates or regenerates a unique ID for an existing installation.

**Note:** The installation ID is not the same as the alias name for an installation.

### Syntax

- Command Central syntax:

```
cc exec landscape nodes node_alias generateNodeId [options]
```

*options*:

```
[{--debug | -d}]
[{--error | -r} file]
[{--log | -l} file]
[{--password | -p} password]
[{--quiet | -q}]
[{--server | -s} url]
[{--username | -u} user_name]
```

- Not applicable to Platform Manager.

## Arguments and Options

Argument or Option	Description
<code>node_alias</code>	<p>Required. Specifies the alias name of the installation for which you want to generate an ID.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<code>[options]</code>	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a>.</p>

## Usage Notes

- Typically, you should not need to generate or regenerate an ID for an installation. Command Central generates an ID for an installation when it is originally added to Command Central management, for example, by executing the [cc create landscape nodes](#) command.

You might regenerate an ID if you have an installation with a duplicate ID. This can occur, for example, if you copy an image of an installation.

- The `cc exec landscape nodes` command stores the newly generated ID in its proper location. The command does not return the new ID as output.

## Example When Executing on Command Central

To generate an ID for the installation with alias name "sag01" using the authorization of the user with user name "Administrator" and password "manage":

```
cc exec landscape nodes sag01 generateNodeId --username Administrator
--password manage
```

Because the `{--server | -s}` option is not specified, the command uses the default server. For more information, see ["--server | -s" on page 77](#).

## Related Commands

[cc create landscape nodes](#)

[cc delete landscape nodes](#)

[cc get landscape nodes](#)

[cc list landscape nodes](#)

[cc update landscape nodes](#)

## cc get landscape nodes

Retrieves information about a specified installation. Information about an installation can include:

- Alias name
- Display name
- Description, or null if none is assigned
- URL of the Command Central that manages the installation
- Status of the Command Central that manages the installation

### Syntax

- Command Central syntax:

```
cc get landscape nodes alias [options]
```

```
options:
[--accept | -a] content_type
[--debug | -d]
[--error | -r] file
[--format | -f] {tsv args | text | xml | csv args | json}
[--log | -l] file
[--output | -o] file
[--password | -p] password
[--quiet | -q]
[--server | -s] url
[--username | -u] user_name
```

- Not applicable to Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	Required. Specifies the alias name of the installation for which you want to retrieve information.  You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a> .
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Usage Notes

- The information for an installation can include the status of the Platform Manager that manages the installation. The status is:

- ONLINE when Command Central can connect with the Platform Manager.
- OFFLINE when Command Central cannot connect to the Platform Manager, for example, if Platform Manager is not running or if there are other connection issues.
- If a Platform Manager is OFFLINE, the command only retrieve the Platform Manager manager status for the installation because the command relies on the Platform Manager to provide the other installation information it retrieves.

### Example When Executing on Command Central

To execute a command on the Command Central server with host name “rubicon” and port “8090” to retrieve information for the installation with alias name “sag01” using the authorization of the user with user name “Administrator” and password “manage”, and have the information returned to the console in JavaScript Object Notation format:

```
cc get landscape nodes sag01 --format json --server http://rubicon:8090/cce
--username Administrator --password manage
```

### Related Commands

[cc create landscape nodes](#)

[cc delete landscape nodes](#)

[cc list landscape nodes](#)

[cc update landscape nodes](#)

## cc list landscape nodes

Lists the installations that Command Central manages. Information about an installation can include:

- Alias name
- Display name
- Description, or null if none is assigned
- URL of the Platform Manager that manages the installation
- Status of the Platform Manager that manages the installation

### Syntax

- Command Central syntax:

```
cc list landscape nodes [node_alias] [options]

options:
[--accept | -a] content_type
[--debug | -d]
[--error | -r] file
[--format | -f] {tsv args | text | xml | csv args | json}
[--log | -l] file
[--output | -o] file
[--password | -p] password
[--quiet | -q]
```

```
[--server | -s} url ]
[--username | -u} user_name ]
```

- Not applicable to Platform Manager.

### Arguments and Options

Argument or Option	Description
[ <i>node_alias</i> ]	Optional. Specifies the alias name of the installation for which you want to list information.
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

- If you do not specify the alias of a specific installation, the command lists installations that Command Central manages.
- The information for an installation can include the status of the Command Central that manages the installation. The status is:
  - ONLINE when Command Central can connect with the Platform Manager
  - OFFLINE when Command Central cannot connect to the Platform Manager, for example, if Platform Manager is not running or if there are other connection issues
- If a Platform Manager is OFFLINE, the command only retrieve the Platform Manager manager status for an installation because the command relies on the Platform Manager to provide the other installation information it retrieves.

### Example When Executing on Command Central

To list all installation that the Command Central with host name "rubicon" and port "8090" manages, using the authorization of the user with user name "Administrator" and password "manage", and have the information returned to the output file "nodelist" in XML format:

```
cc list landscape nodes --format xml --output nodelist
--server http://rubicon:8090/cce --username Administrator
--password manage
```

### Related Commands

[cc create landscape nodes](#)

[cc delete landscape nodes](#)

[cc get landscape nodes](#)

[cc update landscape nodes](#)

## cc update landscape nodes

Updates the properties assigned to an installation, for example, the display name or description.

### Syntax

- Command Central syntax:

- To specify the updated data for the landscape on the command line:

```
cc update landscape nodes node_alias [name=name]
[description=description] [options]
```

- To specify the updated data for the landscape in an input data file:

```
cc create landscape nodes node_alias
{--input | -i} filename{.xml|.json} [options]
```

```
options:
[--debug | -d]
[--error | -r] file
[--log | -l] file
[--media-type | -m] content-type
[--password | -p] password
[--quiet | -q]
[--server | -s] url
[--username | -u] user_name
```

- Not applicable to Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	Required. Specifies the alias name of the installation whose description you want to update.  You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a> .
[name= <i>name</i> ]	Optional. Specifies the updated display name for the installation.  If you use a value that includes spaces, place quotes around the value, for example:  name="My installation"

Argument or Option	Description
<code>[description=<i>description</i>]</code>	<p>Optional. Specifies the updated description for the installation.</p> <p>If you use a value that includes spaces, place quotes around the value, for example:</p> <pre>description="A description with spaces"</pre>
<code>[{--input   -i} filename{.xml .json}]</code>	<p>Optional. Identifies an input file that contains the updated data for the landscape. For more information, see "<a href="#">--input   -i</a>" on page 70.</p> <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p><b>Tip:</b> To determine how to specify the data in the input file, use the <a href="#">cc get landscape nodes</a> to retrieve data for the node you want to update. For example, if you want to use an XML input file, use <a href="#">cc get landscape nodes</a> with the <code>--format xml</code> option to retrieve the data in XML format.</p> </div>
<code>[options]</code>	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see "<a href="#">Options for the Commands</a>" on page 59.</p>

### Usage Notes

- You must specify at least one of the `name` or `description` arguments to indicate the item that you want to update for the installation.

### Example When Executing on Command Central

To update the installation with alias name "sag01" to use the description, "updated version":

```
cc update landscape nodes sag01 description="updated version"
--password "secret"
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see "[--server | -s](#)" on page 77 and "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.

### Related Commands

[cc create landscape nodes](#)

[cc delete landscape nodes](#)

[cc get landscape nodes](#)

[cc list landscape nodes](#)

## Using Command Central to Manage Landscapes with Lower Versions

---

### About Managing Landscapes with Lower Versions

You can use the latest version of Command Central to manage landscapes with lower versions. For example, you can use Command Central version 9.8 to manage a landscape with version 9.7.

The sections in this topic describe how Command Central performs some operations and functions when you use the latest version of Command Central to create or manage a landscape with a lower version.

### Applying Snapshot Templates

When applying a snapshot template on Platform Manager 9.7 or lower, you must consider the following:

- Some products require that you apply a snapshot template in several consecutive steps. For example, when you are applying an Integration Server snapshot template that contains products, fixes, and configuration, you must:
  1. Apply the products.
  2. Restart the *IS\_instancename* component, where *instancename* is the name of the Integration Server instance, and verify that its status is **Online**.
  3. Apply the fixes.
  4. Repeat the second step.
  5. Apply the configuration.

**Note:** Software AG recommends restarting the *IS\_instancename* component after applying a configuration.

Generally, when applying a snapshot template configuration against Platform Manager version 9.7 or lower, the template will be applied successfully on product components that are running. However, some products, for example OSGI-CTP, do not require that the product is running to apply fixes or configuration. With these products you can apply a snapshot template with products, fixes, and configuration in one step.

- You must first install the products and fixes, start the product components, and then apply the template.

- Command Central does not automatically restart product components. You must restart product components that require restart to use configuration changes manually. The apply template command does not automatically restart components.

## **Configuring Authentication Credentials**

When you use the latest version of Command Central to manage installations running with Platform Manager version 9.7 or lower, you can create and update the authentication credentials for run-time components. However, you can not retrieve or view the updated credentials using the Command Central web user interface or the get security credentials command in the command line tool.



---

# 5 Securing the Command Central Landscape

---

■ About Securing the Command Central Landscape .....	148
■ Setting Up the Administrator User Password for Command Central .....	148
■ Setting Up the Administrator User Password for all Platform Managers .....	149
■ Setting Outbound Authentication .....	150
■ Verifying the Outbound Authentication Settings .....	155
■ Accessing Administrative Interfaces through Command Central .....	155
■ Working with Configurations that Include Passwords .....	156
■ Security Credentials Commands .....	158
■ Securing Communication with the Command Central Server .....	166

## About Securing the Command Central Landscape

---

When securing the Command Central landscape, you must consider user access to Command Central and Platform Manager, communication between Command Central and Platform Manager and the Software AG product installations, and external access to Command Central and Platform Manager. This chapter addresses the tasks that must be implemented to secure these interactions.

## Setting Up the Administrator User Password for Command Central

---

After installing Command Central, Software AG recommends you perform the following tasks to set up security.

---

### To set up secure communication with Command Central

1. In Command Central, configure the new user credentials for the Command Central instance as follows:
  - a. In the Environments pane, select the environment that contains the Command Central instance.
  - b. In the Instances table select the **CCE** instance.
  - c. On the Configuration tab, select **Users**.
  - d. On the Users page, click **Administrator** and edit the user credentials.

To verify the new credentials, re-start the browser and log on with the new Administrator password.

2. Update the Command Central command line tool configuration file:
  - a. Execute a command, for example, `cc list landscape nodes`. The command returns `ERROR 401`.
  - b. In a text editor, open the `$HOME/.sag/cc.properties` file and specify the new password in the `password` property.

To verify the new credentials, execute the same command again, for example `cc list landscape nodes`. The command executes successfully.

Repeat the steps in this procedure to change the default passwords for the Guest, Operator, and Viewer users.

## Using a Unix Shell Script to Change the Administrator Password for Command Central

You can use the following sample Unix shell script to change the Administrator user password for Command Central.

```
NODE_ALIAS=local
USERNAME=Administrator
PASSWORD=manage123
RCID=OSGI-CCE

cc get configuration data $NODE_ALIAS $RCID COMMON-LOCAL-USERS
-Administrator -o administrator.xml
sed "s,/>, ><Password>${PASSWORD}</Password>
</User>,g" administrator.xml > administrator_new.xml

cc update configuration data $NODE_ALIAS $RCID COMMON-LOCAL-USERS
-Administrator -o administrator_new.xml

sed "s,^password=.*$,password=${PASSWORD},g" $HOME/.sag/cc.properties >
cc.properties
cp cc.properties $HOME/.sag/cc.properties
cc get landscape nodes
```

## Setting Up the Administrator User Password for all Platform Managers

After installing Platform Manager, Software AG recommends you change the default Platform Manager administrator password.

You can specify a different password for each installed Platform Manager. The password you specify for Platform Manager can differ from the Command Central password.

### To set up secure communication with Platform Manager

1. In the Environments pane in Command Central, select the environment that contains the Platform Manager instance.
2. In the Instances table, select the **SPM** instance.
3. On the Configuration tab, select **Users**.
4. On the Users page, click **Administrator**.
5. Click **Edit** and specify the new password for **Administrator**.

To verify the new credentials click  and check the monitoring KPIs for the SPM component.

## Using a Unix Shell Script to Change the Administrator Password for Platform Manager

You can use the following sample Unix shell script to change the Administrator user password for Platform Manager:

```
NODE_ALIAS=local
PASSWORD=secret
RCID=OSGI-SPM

cc get configuration data $NODE_ALIAS $RCID COMMON-LOCAL-USERS
-Administrator -o administrator.xml
sed "s,/,>><Password>${PASSWORD}</Password>
</User>,g" administrator.xml > administrator_new.xml
cc update configuration data $NODE_ALIAS $RCID COMMON-LOCAL-USERS
-Administrator -o administrator_new.xml
# verify connection
cc list inventory components nodeAlias=$NODE_ALIAS -e $RCID
-w 0 refresh=true
```

## Setting Outbound Authentication

As part of its normal operations, Command Central might connect to applications and subsystems such as Integration Servers, Brokers, and My webMethods Server. By default Command Central uses trusted authentication when administering run-time components in OSGI profiles with version 9.8 or higher. With trusted authentication, Command Central is not required to supply authentication credentials, such as user name and password, to each of these systems before connecting to them.

You can change the authentication mode for run-time components to basic authentication. With basic authentication, Command Central uses the basic credentials to identify itself or *authenticate* to the other systems. When you configure Command Central to connect to a managed runtime, you specify the password Command Central must send to the managed runtime to connect to it. Command Central sends the configured basic authentication credentials to the Platform Manager installation that hosts the run-time component. The Platform Manager installation should be online and accessible.

To use basic authentication, you must change the authentication mode for a run-time component to **Fixed User**. For more information about changing the authentication mode, see ["Changing Authentication Mode" on page 151](#).

Command Central uses basic authentication with a fixed user to communicate with Platform Manager. With **Fixed User** authentication, the authentication credentials for the Platform Manager will be fixed.

Outbound authentication does not authorize access to resources.

**Note:** For information about the limitations when using Command Central to manage run-time components on Platform Manager installations with earlier versions, see "[Configuring Authentication Credentials](#)" on page 145.

## Changing Authentication Mode

By default, the Authentication mode for run-time components that support trusted authentication is set to **Trusted**. In the instance **Overview** tab, click  in the **Authentication** field to change the authentication mode using the Authentication Mode dialog box.

When you specify the authentication mode for an instance, that authentication mode is also set for all layered product instances of the main product instance. For example, Business Rules uses the authentication mode set for the Integration Server instance to which it belongs. However, changing the authentication mode for the OSGI profile of Integration Server does not change the authentication mode for the Integration Server run-time component that belongs to that OSGI profile.

## Changing the Administrator User Password for Managed Products in Command Central

You change the Administrator password for a product managed by Command Central in the Command Central web user interface. After changing the Administrator password for a managed product in Command Central, the outbound credentials are updated automatically.

### To change the Administrator user password for a product in Command Central

1. In the Environments pane in Command Central, select the environment that contains the managed product instance.
2. In the Instances table, select the product component for which you want to change the user credentials as follows:

<u>For this product...</u>	<u>Select...</u>
Integration Server	Integration Server component under <code>integrationServer-<i>instancename</i></code> , for example <code>integrationServer-default</code>
My webMethods Server	My webMethods Server component under <code>MWS-<i>instancename</i></code> , for example <code>MWS-default</code> .
CentraSiteSoftware AG Runtime	CTP

For this product...	Select...
Infrastructure Data Collector	InfraDC <b>Note:</b> The OSGI-InfraDC profile uses the same users that are managed through the OSGI-CTP profile.
Universal Messaging	Universal-Messaging- <i>instancename</i> , for example Universal-Messaging-nirvana <b>Note:</b> By default Universal Messaging does not use basic authentication. Before you can use Command Central to manage Universal Messaging users, you must configure Universal Messaging to use basic authentication. For more information, see the Universal Messaging documentation.

- On the Configuration tab, select **Users**.
- On the Users page, click **Administrator**.
- Click **Edit** and specify the new password for **Administrator**.

To verify the new credentials click  and check the monitoring KPIs for the product component.

## Changing the Administrator User Password for Managed Products in the Product's User Interface

You can change the Administrator user password for a product managed by Command Central using the product's user interface. In this case, after you change the Administrator user password in the product's user interface, you must update the outbound credentials for the managed product in Command Central. The following table lists the product user interfaces that you use to change the Administrator user password.

Product	Action
Integration Server	In Integration Server Administrator, change the Administrator password or create a new user who is a member of the Administrator group on the server instance. For more information, see <i>webMethods Integration Server Administrator's Guide</i> .

Product	Action
My webMethods Server	In My webMethods, change the Administrator and sysadmin passwords. For more information, see <i>Administering My webMethods Server</i> .
CentraSiteSoftware AG RuntimeInfrastructure Data Collector	Using the internaluserrepo script, change the Administrator user password in the <i>Software AG_directory/common/conf/users.txt</i> file. For more information about how to use the internaluserrepo script, see " <a href="#">internaluserrepo Script</a> " on page 171.
Universal Messaging	Using the internaluserrepo script, change the Administrator user password in the <i>Software AG_directory/UniversalMessaging/server/umserver_instance/users.txt</i> file. For more information about how to use the internaluserrepo script, see " <a href="#">internaluserrepo Script</a> " on page 171.

**Note:** By default Universal Messaging does not use basic authentication. Before you can use Command Central to manage Universal Messaging users, you must configure Universal Messaging to use basic authentication. For more information, see the Universal Messaging documentation.

After changing the Administrator user password using the product's user interface, in Command Central, use the following steps to update the Administrator user password for a product component with the new password:

**Note:** You can also configure credentials for the product components using the `cc update configuration data` command.

#### To update the Administrator user password for managed products

1. In the Environments pane, select the environment that contains the product instance.
2. In the Instances table, select the instance of the product component for which you want to change the user credentials as follows:

For this product...	Select...
Integration Server	<code>integrationServer-instancename</code> , for example <code>integrationServer-default</code> , and its child Integration Server component.

<u>For this product...</u>	<u>Select...</u>
My webMethods Server	MwsProgramFiles- <i>instancename</i> , for example MWS-default, and the My webMethods Server component.
CentraSiteSoftware AG Runtime	OSGI-CTP
Infrastructure Data Collector	OSGI-InfraDC
	<b>Note:</b> The OSGI-InfraDC profile uses the same users that are managed through the OSGI-CTP profile.
Universal Messaging	Universal-Messaging- <i>instancename</i> , for example Universal-Messaging-nirvana

3. On the Overview tab, in the Details pane, click  in the **Authentication** field.
4. In the Authentication Mode dialog box, select **Fixed User**.
5. Enter the new **User Name** and **Password**.

**Note:** Command Central cannot manage a Broker Server that has authentication enabled. If you want to manage a Broker Server using Command Central, you must disable their authentication feature. The only **Authentication Mode** that Command Central supports for Broker is **None**.

For Universal Messaging, the default Authentication Mode in Command Central is **None**. You can set the Universal Messaging outbound authentication to **Fixed User** when basic authentication is enabled on the Universal Messaging server and at least one user is configured.

## Using Unix Shell Scripts to Change Connection Credentials for Managed Products

You can use the following sample Unix shell script to configure basic authentication credentials for product components managed by Command Central.

```

NODE_ALIAS=local
USERNAME=Administrator
PASSWORD=secret
RCID=integrationServer-default
# RCID=MwsProgramFiles-default
# RCID=Universal-Messaging-nirvana
# RCID=OSGI-CTP
# RCID=OSGI-InfraDC

cc get configuration data $NODE_ALIAS $RCID COMMON-LOCAL-USERS-Administrator

```

```

-o administrator.xml
sed "s,/>, ><Password>${PASSWORD}</Password></User>,g" administrator.xml >
  administrator_new.xml
cc update configuration data $NODE_ALIAS $RCID COMMON-LOCAL-USERS
-Administrator -o administrator_new.xml

# verify connection
cc get monitoring runtimestatus $NODE_ALIAS $RCID -e ONLINE

```

## Verifying the Outbound Authentication Settings

Use the following steps to verify that Command Central is configured with the correct outbound authentication settings.

### To verify that Command Central is configured with the correct user credentials

1. In Command Central, on the Overview tab for the product component, click . Check that the product status is **Online** and the JVM KPIs are updated.
2. On the Logs tab, check the product log for authentication errors.

## Accessing Administrative Interfaces through Command Central

In Command Central, single sign-on (SSO) is designed to manage webMethods products using an administrative link without any post-installation configuration. When performing advanced configuration tasks, you might need to access the product's primary administrative interface. Command Central provides a link to the administrative interface on the Instances Overview page for each managed product. For example, when you click the Integration Server link on the Overview page of an Integration Server instance, Command Central redirects the browser to the corresponding Integration Server Administrator URL.

You must have administrative credentials to access the Command Central web user interface's administration links to Integration Server and Broker Server.

**Important:** Use *only* one Command Central instance to manage a landscape. You cannot access the Command Central web user interface for a Command Central instance from another Command Central instance.

Use Enterprise Manager to perform advanced configuration tasks of Universal Messaging. You cannot access Enterprise Manager through Command Central.

For information about generating and configuring custom SSO and SAML certificates for Software AG Common Platform-based products, see the Software AG Security Infrastructure documentation.

## Working with Configurations that Include Passwords

When you want to retrieve or update run-time component configuration settings that contain passwords, you can provide a shared secret that Platform Manager uses to create a security context and encrypt the passwords included in the configuration data you want to retrieve.

Using a shared secret is also useful when creating a template that contains configuration. When you set a shared secret for a template, any configuration passwords included in the template .zip archive are encrypted and you can securely apply the template to create new installations.

The following table lists the types of shared secret that you can specify in Command Central and where to find information about how to define a shared secret using the Command Central web user interface or command line interface.

Type of shared secret	Define this shared secret for...	For more information, see...
Global	The landscape	<ul style="list-style-type: none"> <li>■ <a href="#">cc add security credentials sharedsecret</a></li> <li>■ <a href="#">"Configuring a Shared Secret Globally or for an Environment" on page 157</a></li> </ul>
Environment	The environment to which the run-time component configuration belongs	<ul style="list-style-type: none"> <li>■ <a href="#">cc add security credentials sharedsecret</a></li> <li>■ <a href="#">"Configuring a Shared Secret Globally or for an Environment" on page 157</a></li> </ul>
Template	A template created or applied in Command Central	<ul style="list-style-type: none"> <li>■ <a href="#">cc create templates snapshot</a></li> <li>■ <a href="#">cc exec templates snapshot apply</a></li> <li>■ <a href="#">"Creating a Snapshot Template" on page 320</a></li> <li>■ <a href="#">"Applying a Snapshot Template" on page 322</a></li> </ul>

Command Central determines which shared secret to use as follows:

1. If you specify a shared secret when creating or applying a template that contains configuration, Command Central uses the template shared secret.
2. If the run-time component runs in an installation that belongs to a single environment, Command Central uses the shared secret specified for that environment.
3. If the run-time component runs in an installation that belongs to several environments or does not belong to an environment, Command Central uses the global shared secret.
4. If you do not define a custom global shared secret, Command Central uses the default shared secret with which Command Central is installed. Software AG recommends that after installing Command Central you should replace the default global shared secret with a custom global shared secret. For more information about configuring a global shared secret, see ["Configuring a Shared Secret Globally or for an Environment" on page 157](#).

## Configuring a Shared Secret Globally or for an Environment

### To specify a shared secret globally or for an environment

1. In Command Central, in the Environments pane, select the environment for which you want to specify a shared secret. To specify a global shared secret, select **ALL**.
2. Click  and select **Configure Shared Secret**.
3. Type a shared secret in the **Shared Secret** field and **Save**.

### Specify Environment and Global Shared Secrets Using Commands

You can specify a global or environment shared secret using the `cc add security credentials sharedsecret` command. For more information about the command, see [cc add security credentials sharedsecret](#).

#### Example Using an Environment Shared Secret

To update the proxy server details for a run-time component with ID "OSGI-SPM" that runs in the installation with alias name "sag01" that belongs to an environment with alias "Test":

1. Configure "secret123" as the shared secret for the "Test" environment":

```
cc add security credentials sharedsecret
environment=Test secret=secret123
```

2. Retrieve the instance data for the configuration instance with ID "COMMON-PROXY" for the "OSGI-SPM" run-time component that runs in the "sag01" installation:

```
cc get configuration data sag01 OSGI-SPM COMMON-PROXY
--output configWithEncryptedData.xml
```

The `configWithEncryptedData.xml` file contains the proxy details including the password, encrypted with the “secret123” shared secret configured for the “Test” environment.

3. Edit the proxy details in the `configWithEncryptedData.xml` file as required without changing the encrypted password.
4. Update the instance data for the configuration instance with ID “COMMON-PROXY”:

```
cc update configuration data sag01 OSGI-SPM COMMON-PROXY
--input configWithEncryptedData.xml
```

Command Central updates the proxy details using the data in the `configEncryptedData.xml` file and the “secret123” shared secret configured for the environment.

## Configuring a Shared Secret for a Template

You configure a shared secret for a template when creating a template in Command Central, as described in ["Creating a Snapshot Template" on page 320](#). When applying the template, you must provide the same shared secret that you specified when creating the template. If you do not specify the correct shared secret, the encrypted data will not be decrypted and Command Central returns an error.

For information about how to use the Command Central command line interface to specify a shared secret for a template, see [cc create templates snapshot](#) and [cc delete templates snapshot](#).

## Security Credentials Commands

### cc add security credentials

Adds security credentials for an installation or run-time component. Command Central does not store security credentials for run-time components. Command Central only stores the credentials it uses to connect to Platform Manager nodes. When adding security credentials for all run-time components, Command Central propagates the credentials to the run-time component installation. Command Central does not propagate *only* the credentials for the connection between Command Central and the Platform Manager installation.

#### Syntax

- Command Central syntax:

- With parameters

```
cc add security credentials nodeAlias=node_alias
[runtimeComponentId=componentid]
authenticationType={NONE|BASIC|TRUSTED} [username=username]
[password=password] [options]
```

### ■ With input file

```
cc add security credentials nodeAlias=node_alias
[runtimeComponentId=componentid] [--input | -i] file{.xml|.json}
[options]
```

```
options:
[--debug | -d]
[--error | -r] file
[--log | -l] file
[--password | -p] password
[--quiet | -q]
[--server | -s] url]
[--username | -u] user_name]
```

### ■ Not supported by Platform Manager

## Arguments and Options

Argument or Option	Description
nodeAlias= <i>node_alias</i>	<p>Required. Specifies the alias name of the installation for which you want to associate the security credentials.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
[runtimeComponentId= <i>componentid</i> ]	<p>Optional. Specifies the run-time component for which you want to associate the security credentials.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>Important:</b> When configuring the credentials used for communication between Command Central and the Platform Manager installation, set this argument as follows: runtimeComponentId=SPM-CONNECTION.</p> </div>
authenticationType={NONE BASIC TRUSTED}	<p>Required when you use the command syntax with parameters. Specifies the authentication type to use for the run-time component. Valid values:</p> <ul style="list-style-type: none"> <li>■ NONE</li> </ul>

Argument or Option	Description
	<ul style="list-style-type: none"> <li>■ BASIC - use basic authentication</li> <li>■ TRUSTED - use trusted authentication</li> </ul>
[username= <i>username</i> ]	Optional. Specifies the user name to use when the authentication type is set to BASIC.
[password= <i>password</i> ]	Optional. Specifies the password to use when the authentication type is set to BASIC.
{--input   -i} <i>file</i> {.xml .json}	<p>Required when you use the command syntax with input file. Identifies an input file that contains the data for the security credentials. For more information, see "<a href="#">--input   -i</a>" on page 70.</p> <div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <p><b>Tip:</b> To determine how to specify the data in the input file, use <code>cc get security credentials</code> to retrieve data for existing security credentials. For example, if you want to use an XML input file, use <code>cc get security credentials</code> with the <code>--format xml</code> option to retrieve the data in XML format.</p> </div>
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Usage Notes

- By default, if you omit an argument, the command applies the credentials to all items. For example, if you omit the `[runtimeComponentId=componentid]` argument, the command applies the credentials to all run-time components.
- You can set different credentials for the Platform Manager servers in your landscape using the Command Central web user interface or the command line tool. For example, you can configure Command Central to connect to one Platform Manager

as “Administrator1/manage1” and the second Platform Manager server as “Administrator2/manage2”. For example:

```
cc add security credentials runtimeComponentId=SPM-CONNECTION
nodeAlias=sag01 --input c:\inputs\creds_data.xml
```

**Note:** This command will not return an error. However, Command Central does not use the supplied credentials for connection to specified Platform Manager on the specified installation.

### Examples When Executing on Command Central

In the following examples, because the {--server | -s} and {--username | -u} options are not specified, the commands use the default server and user name. For more information, see “--server | -s” on page 77 and “--username | -u” on page 80. The commands specify “secret” for the user’s password.

- To update the security credentials used for the communication between Command Central and the Platform Manager installed in the installation with alias name “sag01”, using authentication type “BASIC”:

```
cc add security credentials nodeAlias=sag01
runtimeComponentId=SPM-CONNECTION
authenticationType=BASIC username=Administrator password=secret
```

- To update the security credentials used for the communication between Command Central and the Platform Manager installed in the installation with alias name “sag01”, using an input file:

- After changing the Command Central default administrator password, create the c:\inputs\credentials\_osgi.xml file. Open the credentials\_osgi.xml file and include the updated security credentials for the run-time component with ID “SPM-CONNECTION”.

- Execute the following command:

```
cc add security credentials nodeAlias=sag01
runtimeComponentId=SPM-CONNECTION --input credentials_osgi.xml -p newpassword
```

- To update the security credentials for a run-time component with ID “MwsProgramFiles” installed in the installation with alias name “sag01”, using trusted authentication:

```
cc add security credentials nodeAlias=sag01
runtimeComponentId=MwsProgramFiles
authenticationType=TRUSTED
```

- Security credentials data is in the c:\inputs\creds\_data.xml file. To add security credentials for all Integration Server run-time components on all installations:

```
cc add security credentials runtimeComponentId=IntegrationServer-instanceName
--input c:\inputs\creds_data.xml --password secret
```

- Security credentials data is in the c:\inputs\creds\_data.xml file. To add security credentials for the Integration Server run-time components on installations with alias names “sag01” and “sag02”, use the following two commands:

```
cc add security credentials nodeAlias=sag01
runtimeComponentId=IntegrationServer-instanceName
--input c:\inputs\creds_data.xml --password secret
```

```
cc add security credentials nodeAlias=sag02
runtimeComponentId=IntegrationServer-instanceName
--input c:\inputs\creds_data.xml --password secret
```

### Related Commands

[cc get security credentials](#)

[cc delete security credentials](#)

## cc add security credentials sharedsecret

Add shared secret for a landscape or an environment.

### Syntax

- Command Central syntax:

```
cc add security credentials sharedsecret [environmentAlias=alias]
secret=text_string [options]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
[environmentAlias= <i>alias</i> ]	Optional. Specify the alias for the environment for which you want to set a shared secret.  If you omit this parameter, the command sets a global shared secret for the whole landscape.
secret= <i>text_string</i>	Specify a shared secret password that Platform Manager uses to encrypt or decrypt configuration passwords.
[options]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Examples When Executing on Command Central

- To configure "mysecret" as the global shared secret for the landscape:

```
cc add security credentials sharedsecret secret=mysecret
```

- To configure "mysecret123" as the shared secret for the environment with alias "env1":

```
cc add security credentials sharedsecret environmentAlias=env1 secret=mysecret123
```

## cc delete security credentials

Deletes security credentials from an installation or run-time component.

### Syntax

- Command Central syntax:

```
cc delete security credentials [nodeAlias=node_alias]
[runtimeComponentId=componentid] [options]

options:
[--debug | -d]
[--error | -r] file
[--force]
[--log | -l] file
[--password | -p] password
[--quiet | -q]
[--server | -s] url]
[--username | -u] user_name]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
[nodeAlias= <i>node_alias</i> ]	Optional. Specifies the alias name of the installation for which you want to delete security credentials.  You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a> .
[runtimeComponentId= <i>componentid</i> ]	Optional. Specifies the run-time component for which you want to delete security credentials.  You can determine the IDs for run-time components using <a href="#">cc list inventory components</a> .
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

## Usage Notes

- By default, if you omit an argument, the command removes the credentials from all items. For example, if you omit the `[runtimeComponentId=componentid]` argument, the command removes the credentials from all run-time components.
- When you remove credentials, Command Central uses the default credentials.

## Examples When Executing on Command Central

- To remove the security credentials for the Integration Server run-time component that is running on the installation with alias name “sag02” using the authorization of the user with user name “Administrator” and password “manage”:

```
cc delete security credentials nodeAlias=sag02
runtimeComponentId=IntegrationServer-instanceName
--username Administrator --password manage
```

After removing the security credentials, the Integration Server uses the default credentials, that is, user name “Administrator” and password “manage”.

- To remove the security credentials for all Integration Server run-time components running on all installations using the authorization of the user with user name “Administrator” and password “manage”:

```
cc delete security credentials
runtimeComponentId=IntegrationServer-instanceName
--username Administrator --password manage
```

## Related Commands

[cc add security credentials](#)

[cc get security credentials](#)

## cc get security credentials

Retrieves security credentials that are associated with an installation or run-time component.

### Syntax

- Command Central syntax:

```
cc get security credentials [nodeAlias=node_alias]
[runtimeComponentId=componentid] [options]
```

```
options :
[--accept | -a] content_type
[--debug | -d]
[--error | -r] file
[--format | -f] {xml | json}
[--log | -l] file
[--output | -o] file
[--password | -p] password
[--quiet | -q]
[--server | -s] url ]]
```

```
[--username | -u} user_name]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
[ <i>nodeAlias=node_alias</i> ]	Optional. Specifies the alias name of the installation for which you want to retrieve security credentials.  You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a> .
[ <i>runtimeComponentId=componentid</i> ]	Optional. Specifies the run-time component for which you want to retrieve security credentials.  You can determine the IDs for run-time components using <a href="#">cc list inventory components</a> .
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Usage Notes

- If you do not specify the `{--format | -f}` option, the default output format is XML.
- By default, if you omit an argument, the command retrieves the credentials from all items. For example, if you omit the `[runtimeComponentId=componentid]` argument, the command retrieves the credentials for all run-time components.
- For security reasons, the command does not return the password.

### Example When Executing on Command Central

To execute a command on the Command Central server with host name "rubicon" and port "8090" to retrieve security credentials for the Integration Server run-time component that is running on the installation with alias name "sag01", using the authorization of the user with user name "Administrator" and password "manage", and have the information displayed on the console in XML format:

```
cc get security credentials nodeAlias=sag01
runtimeComponentId=IntegrationServer-instanceName --format xml
--server http://rubicon:8090/cce --username Administrator
--password manage
```

### Related Commands

[cc add security credentials](#)

[cc delete security credentials](#)

## Securing Communication with the Command Central Server

---

### Prerequisites to Configuring a Port for SSL

Command Central provides a default keystore and truststore that are available after installing Command Central. The keystore and truststore are files that function as repositories for storage of keys and certificates necessary for Secure Socket Layer (SSL) authentication, encryption/decryption, and digital signing/verification services. The default keystore and truststore contain signed certification authority (CA) certificates that the Command Central server uses to validate client certificates.

You can replace the Command Central default keystore and truststore files with custom files. For information about creating keystores and truststores, importing keys and certificates into keystores and truststores, and other operations with these files, refer to the documentation for your certificate management tool.

Before configuring an HTTPS port, you must configure the Command Central server to use SSL, using the Command Line tool.

### Configuring SSL Using Command Options

You can secure connections to the Command Central server by using Command Line interface options that contain SSL configuration settings, such as the location of the keystore and truststore files. For more information about the SSL-related command options, see "[Options for the Commands](#)" on page 59.

### Configuring SSL Using Configuration Properties Files

Command Central comes with a default configuration properties file that contains the SSL command options with the default SSL settings. When you have set the `CC_CLI_HOME` and the `PATH` environment variables, the default configuration settings in the `CC_CLI_HOME\conf\cc.properties` file are used by the local Command Central server to communicate over HTTPS.

You should make a copy of the default configuration properties file to create a custom file in which you set all SSL-related configuration settings. Using the `--configuration-file` command option, you specify the location of the custom configuration properties file. For more information about the command option, see "[--configuration-file](#)" on page 61.

## Creating a Custom Command Central Properties File

**Important:** Software AG does not recommend editing or changing the settings in the default `cc.properties` file, located in the `Software AG_directory\CommandCentral\client\conf` directory. Use the following procedure to create a custom configuration properties file and set the required authentication settings in the custom file.

### To create a custom properties configuration file

1. Go to `Software AG_directory\CommandCentral\client\conf`
2. Copy the `cc.properties` files to the following location: `user_home\.sag\cc.properties`

**Note:** To create the `.sag` directory in Windows, at the command prompt type  
`mkdir %HOME%\.sag`

3. Set file permissions for your copy of the `cc.properties` file to prevent other users from accessing the file.
4. Edit the custom `cc.properties` file in a text editor as required.

The following table lists the default configuration settings and the option that you can use to override the default value:

Property	Default Value	Use this option to override the default setting
server	<code>https://localhost:8091/cce</code>	<code>--server</code>
username	Administrator	<code>--user</code>
password		<code>--password</code>
ssl-truststore-file	<code>demo-truststore.jks</code>	<code>--ssl-truststore-file</code>
ssl-truststore-password		<code>--ssl-truststore-password</code>
ssl-trust-all-hosts		<code>--ssl-trust-all-hosts</code>

For more information about the command options listed in the table, see ["Options for the Commands"](#) on page 59.

5. Save the custom `cc.properties` file.

## Configuring SSL Using Scripts

You can secure connections to the Command Central server by using ANT properties. For more information about the SSL-related ANT properties, see ["Parameters to Use with the `ccsetup` Task"](#) on page 82.

## Considerations When Using Configuration Properties

### Determining the Value for Configuration Properties

The following lists the order used to determine the value for any of the configuration properties when executing a CLI command:

1. Value in the first command option or ANT property.
2. Value in the custom `cc.properties` file in the `user_home \.sag\cc.properties` directory.
3. Value in the default `cc.properties` file in the `CC_CLI_HOME\conf` directory if you have set the `CC_CLI_HOME` and the `PATH` environment variables.

### Specifying the Password

When executing a command using the Command Line interface, Command Central prompts for a password each time the `cc` command is executed. You must specify a password for your user and truststore in one of the following:

- The `--password` and `--ssl-truststore-password` options
- The default or custom `cc.properties` configuration files
- The `CC_PASSWORD` environment variable

# 6 Managing Users, Groups, and Roles

---

- Managing Command Central Users, Groups, and Roles ..... 170
- Managing Internally Defined Users in Command Central ..... 170
- Using Internally Defined User and Group Information ..... 171
- Using Externally Defined User and Group Information ..... 175
- Using JAAS with Command Central ..... 180
- Groups ..... 183
- Roles ..... 184

## Managing Command Central Users, Groups, and Roles

---

Command Central uses user, role, and group information to authenticate users and determine the resources a user is allowed to access. This information is stored in the internal user repository.

A *user* is defined by a user name or user ID. Users can be members of groups. They can also be assigned roles within the repository.

A *group* is a defined collection of users. Command Central and Platform Manager support groups as a way to manage users. A user does not have to be a member of any group, but a user can be a member of more than one group. For more information about groups, see ["Groups" on page 183](#).

A *role* is a defined collection of privileges within Command Central. A role consists of specific access control rights or *permissions*. For more information about roles, see ["Roles" on page 184](#).

Roles are assigned to individual users and to groups. When a role is assigned to a group, all members of the group inherit that role. The roles assigned to a user control what permissions the user has when using Command Central and Platform Manager.

After installing Command Central, configure the internal user repository with new users, roles, and groups by adding users, adding groups and assigning users to them, and mapping users to roles and roles to groups. You configure internal users by using the Command Central web user interface or the Command Central command line configuration data commands.

For information about using LDAP and external repositories, see ["Using Externally Defined User and Group Information" on page 175](#).

## Managing Internally Defined Users in Command Central

---

You can use the Command Central user interface to manage internal users for Command Central, Platform Manager, and managed products that support the COMMON-LOCAL-USERS configuration, such as Integration Server, My webMethods Server, and Universal Messaging. To add, update, or remove internal users or change passwords in Command Central, locate the **Users** configuration type as follows:

<b>For</b>	<b>Go to...</b>
Command Central	CCE instance
Platform Manager	SPM instance

For	Go to...
Managed products	The run-time component under the product's OSGI instance

For detailed steps about how to work with configuration types, see "[Configuring Instances](#)" on page 242.

## Using Internally Defined User and Group Information

Command Central can authenticate users against information in the Command Central or Platform Manager internal user repositories, using the users.txt file located in the following directories:

- For Command Central: *Software AG\_directory*\profiles\CCE\configuration\security
- For Platform Manager: *Software AG\_directory*\profiles\SPM\configuration\security

When managing users within the internal user repository, you should assign users and groups to roles in the roles.txt file, and assign users to groups in the groups.txt files. You can find the roles.txt and the groups.txt files in the following directories:

- For Command Central: *Software AG\_directory*\profiles\CCE\configuration\security
- For Platform Manager: *Software AG\_directory*\profiles\SPM\configuration\security

For more information, see "[Groups](#)" on page 183 and "[Roles](#)" on page 184.

**Note:** Immediately after logging on to Command Central for the first time, the administrator should change the default administrator password. For more information see "[Setting Up the Administrator User Password for Command Central](#)" on page 148.

## internaluserrepo Script

You can use the internaluserrepo script when you want to modify the user.txt file for Command Central, Platform Manager, or the Software AG Runtime. For all products managed by Command Central, use the Command Central web user interface or the Command Central command line tool. The internaluserrepo.bat/sh script creates or modifies the users.txt file, adds and deletes users in the file, and changes specified internal user passwords.

The internaluserrepo.bat/sh script is located in the following directory:

*Software AG\_directory*\common\bin

To use internaluserrepo.bat/sh, open a command prompt or console and change the directory to the internaluserrepo.bat/sh script's location.

## Syntax

At the command prompt, use the following syntax:

```
internaluserrepo.bat/sh [-f filename] [-c] [-p password] [-d | -e] userId
```

When the command syntax is not correct, `internaluserrepo.bat/sh` reports an exit status code. When the command syntax is correct, the command prompt returns without any additional information. For more information about exit codes, see ["internaluserrepo Exit Codes" on page 173](#).

## Arguments

The following table provides descriptions for the arguments that can be made to the `internaluserrepo.bat/sh` script.

Argument/Parameter	Description
<code>userId</code>	<p>Required. Creates a new user when the <code>users.txt</code> file exists in the <code>Software AG_directory\profiles\CCE   SPM\configuration\security</code> directory. The specified user ID is added to the file and the <code>internaluserrepo</code> script prompts you for the new user's password.</p> <p>When the specified user ID exists in the <code>users.txt</code> file, the password is changed for that user ID.</p> <p>You can use up to 128 characters for <code>userId</code>.</p> <p>The following are valid characters for a user ID:  <code>[a-z] [A-Z] [0-9] !()-.?[]@_~</code></p>
<code>[-f filename]</code>	<p>Optional. Specifies the location and file name followed by the URL, or the path and name of the file to create.</p>
<code>[-c]</code>	<p>Optional. Creates a <code>users.txt</code> file in the directory where the command is executed when no other options are used.</p>
<code>[-p password]</code>	<p>Optional. Specifies the password.</p> <p>You can use up to 128 characters for the password.</p> <p>The following are valid characters for a password:  <code>[a-z] [A-Z] [0-9] !()-.?[]@_~</code></p>
<code>[-d]</code>	<p>Optional. Specifies the user to delete from the <code>users.txt</code> file.</p>

Argument/Parameter	Description
[-e]	Optional. Checks whether a specified user exists in the users.txt file.  When using option -e, you must also specify the file name and the file's URL using option -f.

## internaluserrepo Exit Codes

The following table describes the exit codes you might encounter when using the `internaluserrepo.bat/sh` script.

Exit Code	Description
-1	The user ID specified with the -e option does not exist in the users.txt file.
1	The password is not set.
2	The user ID is too long.
3	The user ID contains invalid characters.
4	The password contains invalid characters.
5	The password is too long.
6	The internal user repository contains multiple versions of the users.txt file.
7	An invalid version of the configuration file exists in the repository.  Supported versions are 2.0 and above.
8	One of the following has occurred: <ul style="list-style-type: none"> <li>■ The file name is not specified in the command and the default file cannot be located.</li> <li>■ The file specified cannot be located. Make sure you have entered the correct path and file name.</li> </ul>

Exit Code	Description
9	The <code>internaluserrepo</code> script cannot open or create the <code>users.txt</code> file.
10	The user ID is missing.
11	The command includes conflicting arguments or invalid parameters.

## Adding Users to the Internal User Repository

In addition to the default user (Administrator) in the `users.txt` file, you can add users, such as Guest, Viewer, and Operator, to the internal user repository by running the `internaluserrepo` script. You must have administrator credentials to add users to the internal user repository.

### To add a user to the internal user repository

1. At the command prompt, type the following command to change the directory:

```
cd common\bin
```

2. Type the following command:

- To add a Command Central user:

```
internaluserrepo.bat -f
../../profiles/CCE/configuration/security/users.txt
-c -p password userId
```

- To add a Platform Manager user:

```
internaluserrepo.bat -f
../../profiles/SPM/configuration/security/users.txt
-c -p password userId
```

For example, to add user Administrator1 with a password of manage1, type:

```
internaluserrepo.bat -f
../../profiles/SPM/configuration/security/users.txt
-c -p manage1 Administrator1
```

**Note:** A user name can be fully qualified, such as, LDAP Distinguished Name.

## Deleting Users from the Internal User Repository

You can delete users from the internal user repository (`users.txt` file).

### To delete a user from the internal user repository

1. At the command prompt, type the following command to change the directory:

```
cd common\bin
```

## 2. Type the following command:

- To delete a Command Central user:

```
internaluserrepo.bat -f
  ../../profiles/CCE/configuration/security/users.txt
  -d userId
```

- To delete a Platform Manager user:

```
internaluserrepo.bat -f
  ../../profiles/SPM/configuration/security/users.txt
  -d userId
```

For example, to delete Administrator1, enter the following command:

```
internaluserrepo.bat -f ../../profiles/CCE/configuration/security/
users.txt -d Administrator1
```

## Using Externally Defined User and Group Information

Command Central can use externally defined information for the same purposes it uses internally-defined user and group information:

- To authenticate clients using user names and passwords
- To control who can configure and manage Command Central

You can set up Command Central to access information from an external directory if your site uses one of the following external directories for user and group information:

- Lightweight Directory Access Protocol (LDAP)
- Microsoft Active Directory (AD) acting as an LDAP server

**Note:** Externally defined user and group information does not replace roles and permissions. To control actions within Command Central, as well as access to data, you still need to set up roles and associate users and groups with those roles to allow or deny access to specific actions.

## How Command Central Authenticates Externally Defined Clients

When Command Central is authenticating a client using user names and passwords, it first attempts to find the user name and password in its internal user repository. If it finds an internally-defined user account for the supplied user name, the server authenticates the client using the internally-defined information. If the supplied password is correct, the server proceeds with the request. If the supplied password is not correct, the server rejects the request.

If the server cannot find an internally-defined user account for the supplied user name, the server accesses the external directory (LDAP) to obtain user name and password information for the client. If it finds an externally defined user account, the server authenticates the client using the externally defined information.

If the server cannot find either an internally or externally defined user account for the user, the server rejects the request.

## Overview of Using LDAP

If your site uses Lightweight Directory Access Protocol (LDAP) for user and group information, you can configure Command Central to obtain user and group information from the external directory. You can configure Command Central to use more than one LDAP directory at a time, allowing Command Central to work with different LDAP directories for users in different locations or different organizations. In addition, you can maintain multiple LDAP directories so that one directory serves as a backup for another.

LDAP protocols are designed to facilitate sharing information about resources on a network. Typically, they are used to store profile information, such as user name and password. You can also use them to store additional information. Command Central uses LDAP for performing external authentication.

Using your existing LDAP information allows you to take advantage of a central repository of user and group information. System administrators can add and remove users from the central location. Users do not need to remember a separate password for webMethods applications; they can use the same user names and passwords that they use for other applications. Remember to use your LDAP tools to administer users or groups stored in an external directory.

## LDAP Profile Properties

When you want to use LDAP or Active Directory as an LDAP server for authentication purposes instead of using the internal user repository, you must update the LDAP profile properties.

The following table describes the profile properties for all LDAP connections. Use this information to help you update the LDAP profile. For information about how to configure LDAP profile properties, see ["Configuring LDAP Profile Properties" on page 179](#).

Property Name	Default Value	Description
alias	None	Optional. Specifies the alias for the LDAP configuration entry. When alias is not specified, its value is set to match the url property.  Use any string of characters as the valid value.
url	None	Optional. Specifies the URL for the LDAP server. If you want to use an SSL connection to the LDAP server, ensure the URL starts with ldaps, and provide the truststore

Property Name	Default Value	Description
		<p>or keystore parameters. Use one of the following formats:</p> <ul style="list-style-type: none"> <li>■ <code>ldap://host:port</code></li> <li>■ <code>ldaps://host:port</code></li> </ul>
<code>prin</code>	None	Optional. Specifies the distinguished name (DN) of the technical user who connects to the LDAP server if anonymous access to the LDAP server is not allowed.
<code>cred</code>	None	Optional. Specifies the password of the technical user who connects to the LDAP server. Use <code>cred</code> with the <code>prin</code> property. Use any string of characters as the valid value.
<code>useaf</code>	false	<p>Indicates if an affix (<code>dnprefix</code> or <code>dnsuffix</code>) is used with the LDAP directory entry's distinguished name (dn).</p> <p>When <code>useaf</code> is set to true, the distinguished name uses affixes. When <code>useaf</code> is set to false, the distinguished name does not use affixes.</p>
<code>dnprefix</code>	None	<p>Optional. Specifies the string prefix to add to the DN user name when performing operations on the LDAP server. To use <code>dnprefix</code>, you must set <code>useaf</code> to true.</p> <p>Use any string of characters as the valid value.</p>
<code>dnsuffix</code>	None	<p>Optional. Specifies the suffix to append to the DN user name when performing operations on the LDAP server. To use <code>dnsuffix</code>, you must set <code>useaf</code> to true.</p> <p>Use any string of characters as the valid value.</p>
<code>usecaching</code>	None	Optional. Indicates if the LDAP framework caches users or groups, or both.

Property Name	Default Value	Description
		Set to true to enable caching all users and groups. Set to false to disable caching LDAP users and groups.
<code>matr</code>	None	<p>Optional. Indicates the type of member search operation that is performed using the value in <code>memberinfoingroups</code>.</p> <p>When <code>memberinfoingroups</code> is set to true, <code>matr</code> points from the group to the users that are members of the group. When <code>memberinfoingroups</code> is set to false, <code>matr</code> points from a user entry to the groups for which the user is a member.</p> <p>Use any string of characters as the valid value for <code>matr</code>.</p>
<code>memberinfoingroups</code>	False	<p>Optional. Indicates if the login module searches for users that are members of a group or searches for the groups for which a user is a member. You can only use <code>memberinfoingroups</code> when the value that is provided in <code>matr</code> is applied to users or groups.</p> <p>When <code>memberinfoingroups</code> is set to true, the login module searches users in a group. When <code>memberinfoingroups</code> is set to false, the login module searches groups for a user.</p>
<code>creategroups</code>	True	<p>Optional. Indicates if the login module extracts groups of the logged-on user from the LDAP server.</p> <p>When <code>creategroups</code> is set to true, the login module extracts the groups of the logged-on user from the LDAP server. When <code>creategroups</code> is set to false, the login module does not extract the groups of the logged-on user from the LDAP server.</p>
<code>gidprop</code>	None	Optional. Specifies the LDAP group attribute.

Property Name	Default Value	Description
		Use any string of characters as the valid value for <code>gidprop</code> .
<code>grouprootdn</code>	None	Optional. Specifies the location from which to start searches for groups.  Use any string of characters as the valid value for <code>grouprootdn</code> .
<code>groupobjclass</code>	group	Optional. Indicates that the found object is a group. The login module uses the <code>groupobjclass</code> when searching for groups.
<code>personobjclass</code>	person	Optional. Indicates that the found object is a person. The login module uses the <code>personobjclass</code> when searching for users.

## Configuring LDAP Profile Properties

When you want to use LDAP or Active Directory as an LDAP server for authentication purposes instead of using the internal user repository, you must update the LDAP profile properties using one of two methods:

- Update the JAAS configuration file directly using the command line interface. This method allows you to configure only one LDAP connection.
- Update the `com.softwareag.security.ldap.pid.properties` file, as described in the following procedure. This method allows you to configure connections to multiple LDAPs.

### To configure LDAP properties using the `ldap.properties` file

1. In a text editor, open the `com.softwareag.security.ldap.pid.properties` file located in the following directories:
  - For Platform Manager:`Software AG_directory\profiles\SPM\configuration\com.softwareag.platform.config.propsloader`
  - For Command Central:`Software AG_directory\profiles\CCE\configuration\com.softwareag.platform.config.propsloader`
2. Update the LDAP properties based on your needs.
3. Save and close the file.

## Using JAAS with Command Central

Java Authorization and Authentication Service (JAAS) provides a standards-based mechanism for deploying custom login modules. Using JAAS, you can write your own custom login module to take over the Command Central authentication process. For more information, see the Developing Login Modules section in the webMethods Suite Security Infrastructure documentation.

By making use of the JAAS framework for extending Java code-based security, you can customize Command Central authentication so that multiple login modules can be called during the authentication process. JAAS allows you to specify:

- The order in which custom login modules are called.
- Whether a login module is required or optional.
- The points at which control can pass from a login module back to the controlling application.

When implementing custom login modules using JAAS, you must:

- Write the login module.
- Configure your login module within the appropriate login context in the JAAS configuration file.

**Note:** A JAAS custom login module deals only with *authentication* of Command Central users. You cannot use JAAS for Command Central *authorization*.

## JAAS Configuration File

The JAAS configuration file controls which login modules to use within a JVM. Command Central configures the JVM to use:

- For Command Central: *Software AG\_directory*\profiles\CCE\configuration\security\jaas.config
- For Platform Manager: *Software AG\_directory*\profiles\SPM\configuration\security\jaas.config

**Note:** Command Central does not use the JAAS configuration file located in the *Software AG\_directory*\profiles\CCE\SPM\configuration directory.

A set of JAAS login modules are grouped into what is termed a *login context*. Within each login context, the login modules are specified with their full name, optional parameters, and a designation of the actions to take based on their success or failure. These designations are classified as REQUIRED, REQUISITE, SUFFICIENT, and OPTIONAL. For the login to succeed, the complete login context must succeed.

The JAAS configuration file lists the:

- Available login contexts
- Login modules that will execute
- Order in which the modules will execute
- Settings that determine which actions to take if a module fails

Following is a portion of the default JAAS configuration file for Command Central.

```
Default {
//SSOS login module for SAML signed assertion validation
//com.softwareag.security.idp.saml.lm.
  SAML1AssertValidatorLoginModule sufficient;
//Internal repository login module (java based)
  com.softwareag.security.jaas.login.internal.InternalLoginModule required
    template_section=INTERNAL
    logCallback=true
    internalRepository="C:/wm/kga/common/conf/users.txt"
    create_group_principal=true
    groupRepositoryPath="C:/opt/common/conf/groups.txt";
//Role repository login module
  com.softwareag.security.authz.store.jaas.login.RoleLoginModule optional
    storage_location="C:/SoftwareAG/common/conf/roles.txt";
//SSOS login module for SAML sign assertion generation
//com.softwareag.security.idp.saml.lm.SAML1AssertIssuerLoginModule optional;
}
```

You can modify the JAAS configuration file to allow authentication against user stores other than the Command Central and Platform Manager internal user stores, for example LDAP or the Software AG Common Platform user store in the *Software AG\_directory*\common\conf\users.txt file. You can modify the JAAS configuration file in the Command Central web user interface or the Command Central command line interface.

## Editing the jaas.config File in Command Central

### To edit the jaas.config file in Command Central

1. In Command Central, on the Configuration tab for **Platform Manager** or **Command Central Server**, select **Authentication**.
2. Click **JAAS Settings**.
3. Click **Edit**.
4. Type your changes in the **Content** field and save.

### Editing the jaas.config File Using the cc update configuration Command

In the Command Central command line tool, run the cc update configuration data command. For more information about how to use the command, see ["cc update configuration data" on page 256](#).

## Configuring the jaas.config File to Use LDAP/AD

The following procedure describes how to configure the jaas.config file to use LDAP/AD.

For more information about the LDAPLoginModule, see the Predefined Login Modules section in the *Software AG Security Infrastructure Documentation*

### To configure the jaas.config file to use LDAP/AD

1. Use a text editor to open the jaas.config file located in the following directory:
  - For Command Central: *Software AG\_directory*\profiles\CCE\configuration\security
  - For Platform Manager: *Software AG\_directory*\profiles\SPM\configuration\security
2. Ensure the InternalLoginModule's resolution is set to optional as follows:
 

```
com.softwareag.security.jaas.login.internal.InternalLoginModule optional
```
3. Replace with the following:

```
Default {
//SSOS login module for SAML signed assertion validation
//com.softwareag.security.idp.saml.lm.
  SAMLAssertValidatorLoginModule sufficient;
//Internal repository login module (java based)
  com.softwareag.security.jaas.login.internal.InternalLoginModule optional
    template_section=INTERNAL
    logCallback=true
    internalRepository="/opt/softwareag/common/conf/users.txt"
  create_group_principal=true
  groupRepositoryPath="/opt/softwareag/common/conf/groups.txt";
  com.softwareag.security.sin.is.ldap.lm.LDAPLoginModule optional
    url="ldap://myldapserver:389"
    prin="CN=user,OU=myuser,DC=ldap,DC=server"
    cred="*****"
    gidprop="CN"
    uidprop="CN"
    usecaching="false"
    userrootdn="DC=my,DC=ldap,DC=server"
    mattr="memberOf"
    memberinfoingroups=false
    grouprootdn="DC=my,DC=ldap,DC=server"
    groupobjclass="group"
    personobjclass="person"
    creategroups=true;
//Role repository login module
  com.softwareag.security.authz.store.jaas.login.RoleLoginModule requisite
    storage_location="/opt/softwareag/common/conf/roles.txt";
//SSOS login module for SAML sign assertion generation
//com.softwareag.security.idp.saml.lm.SAMLAssertIssuerLoginModule optional;
};
/*
 * Login context, used in common Platform for management channel .
 */
PlatformManagement {
  //SSOS login module for SAML signed assertion validation
  //used for delegated out only for JMX
```

```

    com.softwareag.security.idp.saml.lm.
    JMXDelegatedAuthLoginModule sufficient;
//Internal repository login module (java based)
    com.softwareag.security.jaas.login.internal.InternalLoginModule required
        template_section=INTERNAL
        logCallback=true
        internalRepository="/opt/softwareag/common/conf/users.txt";
//Role repository login module
    com.softwareag.security.authz.store.jaas.login.RoleLoginModule optional
        storage_location="/opt/softwareag/common/conf/roles.txt";
}

```

4. Save and close the file.

## Groups

A *group* is a defined collection of users. Groups reflect organizational structure, for example, departments within the organization. When a group is assigned a particular role, all members of the group inherit the permissions granted by this role.

Using groups is optional. Use the internal repository groups and users only when LDAP/AD is not used or is unavailable.

When using LDAP/AD, groups can be nested within other groups. However, members of nested groups do not inherit the parent group's roles and their assigned permissions. You must be a direct member of a group to inherit a role and its permissions.

## Default Groups

Command Central comes with the following default groups:

- Administrators, with default user, Administrator
- Viewers, with default user Viewer
- Operators, with default user Operator

## Managing Groups

Groups are managed in the groups.txt file. You can use the Command Central web user interface to edit the groups.txt file for Command Central and Platform Manager.

### To edit the groups.txt file in Command Central

1. In Command Central, on the Configuration tab for **Platform Manager** or **Command Central Server**, select **Groups**.
2. Click **Internal User Groups**.
3. Click **Edit**.
4. Type your changes in the **Content** field and save.

## Related Topics

[Configuring Instances](#)

[Configuration Commands](#)

## Editing the groups.txt File in a Text Editor

Using a text editor, you can add, modify, or delete groups.

### To add, modify, or delete groups in the groups.txt file

1. Use a text editor to open the groups.txt file located in the following directory:
  - For Command Central: *Software AG\_directory*\profiles\CCE\configuration\security
  - For Platform Manager: *Software AG\_directory*\profiles\SPM\configuration\security

2. To add a new group, create a new line and use the following format:

```
group_name:unique_ID :user_name1,user_name2,...user_NameN
```

where *unique ID* is a unique identifier for the group. For example:

```
Administrators:1:Administrator
```

**Note:** A group name can be fully qualified, such as, LDAP Distinguished Name.

3. To modify an existing group, edit the group's information, as needed.
4. To delete a group, delete the group's line in the file.

**Note:** You can also comment out a group's information by including an asterisk (\*) as the first character. For example:

```
*\Group1:25:Operator
```

5. Save and close the file.

## Roles

A *role* is a collection of access control rights or *permissions* within Command Central. Roles are assigned to individual users and to groups. When a role is assigned to a group, all members of the group inherit that role. The roles assigned to a user control what permissions the user has when using Command Central and Platform Manager.

Permissions are managed in the roles.txt file. They are assigned to users at run time. Permissions can manage multiple levels of access (for example, installation node, service, resource type, instance) and different actions (for example, create, read, update, delete).

The following table describes the permissions that can be assigned to roles.

Permission	Description
canread	Provides read-only access to all information, including lists of products, components, configuration, and monitoring data.
canwrite	Provides the ability to create, update, delete all information managed by Command Central and Platform Manager, including nodes, environments, and product configuration.
canexecute	Provides the ability to execute lifecycle operations: start, stop, debug, and restart.

In the `roles.txt` file, you assign permissions to the roles, and map the roles to users or groups. Users are granted permissions based on the group to which they are a member. For more information, see ["Managing Roles" on page 185](#) and ["Groups" on page 183](#).

**Note:** While it is possible to map users to roles, Software AG recommends mapping groups to roles instead. Mapping groups to roles simplifies the authorization model maintenance. Define your authorization model once, and do not implement changes. The only change that should occur in production is mapping users to groups, which is normally done when LDAP/AD is implemented.

## Default Roles

By default, Command Central supports the following roles and their corresponding permissions. For information about adding roles of your own, see ["Managing Roles" on page 185](#).

Role	Permissions
readonlyadmin	canread, canexecute
superadmin	canread, canwrite, canexecute
viewer	canread

## Managing Roles

Roles are managed in the `roles.txt` file. You can use the Command Central web user interface to edit the `roles.txt` file.

**Note:** The users and groups in the roles.txt file must match the users and groups in the user repository. For more information, see ["Groups" on page 183](#) and ["Managing Groups" on page 183](#).

The following sample illustrates a roles.txt file. The table following the sample describes each section of the file. The examples used in the procedure correspond to the sample.

```
[permissions]
permissions:allow=canwrite,canexecute,canread
[roles]role:superadmin=*
role:readonlyadmin=canread,canexecute
role:viewer=canread
[users]
user:"Administrator"=superadmin
[groups]
groups:"Administrators"=superadmin
```

Section	Description
permissions	Lists the permissions that are allowed, such as canread.
roles	Defines roles and the permissions assigned to them. An asterisk (*) denotes all permissions are assigned to the role.
users	Maps users from the users.txt file to roles.
groups	Maps roles to groups defined in the groups.txt file.

#### To edit the roles.txt file in Command Central

1. In Command Central, on the Configuration tab for **Platform Manager** or **Command Central Server**, select **Roles**.
2. Click **Access Roles and Permissions**.
3. Click **Edit**.
4. Type your changes in the **Content** field and save.

#### Related Topics

[Configuring Instances](#)

[Configuration Commands](#)

#### Editing the roles.txt File in a Text Editor

Using a text editor, you can add roles, set permissions for each role, and map roles to groups. Also, you can modify or delete roles.

---

**To add, modify, and delete roles and permissions to the roles.txt file**

1. Use a text editor to open the roles.txt file located in the following directory:
  - For Command Central: *Software AG\_directory*\profiles\CCE\configuration\security
  - For Platform Manager: *Software AG\_directory*\profiles\SPM\configuration\security
2. To add a new role, create a new line in the `roles` section using the following format:

```
role:rolename=permissions
```

For example, to add a new role, superadmin, that has permission to do everything, enter:

```
role:superadmin=*
```

3. To modify an existing role, edit the role's information, as needed.
4. To delete a role, delete the role's line in the file.

**Note:** You can also comment out a role's information by including an asterisk (\*) as the first character. For example:

```
*\Group1:25:Operator
```

5. In the `users` section, map the user from the users.txt file to the new role using the following format:

```
user:"user_name "=role
```

For example, to map the user, Administrator, to the superadmin role, specify the following:

```
user:"Administrator "=superadmin
```

6. In the `groups` section, map the role to a group defined in the groups.txt file, using the following format:

```
group:"group_name "=role name
```

7. Save and close the file.



---

# 7 Viewing Product Inventory

---

■ About Inventory Management .....	190
■ Viewing Products in an Installation .....	190
■ Viewing Fixes Applied to Products in an Installation .....	190
■ Comparing Product Versions, Fixes, and Configurations .....	190
■ Inventory Commands .....	192
■ Resources Commands .....	219

## About Inventory Management

---

Command Central queries the Platform Manager for information about the installed products, versions, and fixes of all the managed products that are part of the installation where the Platform Manager is installed.

## Viewing Products in an Installation

---

You can view information about the products, versions, and the components of the products installed in different installations. Use the following procedure to view the details of the products installed in an installation.

### To view the products in an installation

1. In the Environments pane, select the environment from which you want to view the products details.
2. Select the **Installations** tab.
3. Click the name of the installation you want to inspect.
4. Select the **Products** tab.

## Viewing Fixes Applied to Products in an Installation

---

Use the following procedure to view the details of fixes applied to products in an installation.

### To view the fixes applied to the products in an installation

1. In the Environments pane, select the environment in which you want to view the fix details.
2. Select the **Installations** tab.
3. Click the name of the installation you want to inspect.
4. Select the **Fixes** tab.

## Comparing Product Versions, Fixes, and Configurations

---

### About Comparing Products

You can compare installed products for a quick view of their versions, fixes, and configuration settings. Specifically, you can compare:

- Versions of products existing in the same installation or in different installations.
- Fixes applied to the products existing in the same installation or in different installations.
- Configuration settings of instances of an installation.

## Comparing Product Versions

When you compare product versions, you can see the version numbers of the products in the selected installations as well as the servers on which the products are installed.

---

### To compare the versions of the products installed

1. In the Environments pane, select the environment for which to compare the version numbers of the products.  
To view the installations of all environments, select **All**.
2. Select the **Installations** tab.
3. In the **Installations** tab, select two or up to a maximum of five installations for which you want to compare the product versions.
4. Click  and select **Compare Products**.

## Comparing Fix Levels

When you compare fix levels, you can see the fixes that are applied to the products in the selected installations.

You can also select an installation and click the **Fixes** tab to see the fix names, who installed the fix, whether the fix was installed using Software AG Update Manager, and the date and time when the fix was installed.

---

### To compare the fixes applied to the installed products

1. In the Environments pane, select the environment for which you want to compare the fix levels.  
To view the installations of all environments, select **All**.
2. Select the **Installations** tab.
3. In the **Installations** tab, select two or up to a maximum of five installations for which you want to compare the details of the fixes applied to the products.
4. Click  and select **Compare Fixes**.

## Comparing Configuration Settings

When you can compare the configuration settings of instances, you can quickly identify if there are any differences in the settings.

### To compare the configuration settings of instances

1. In the Environments pane, select the environment for which you want to compare the product configuration settings.  
To view the instances of all environments, select **All**.
2. Select the **Instances** tab.
3. In the **Instances** tab, select two, or up to a maximum of five, instances for comparison.
4. Click  and select **Compare Configuration**.
5. In the drop-down list, select the configuration type (port or license) that you want to compare.
6. Click  to return to the **Instances** view.

## Inventory Commands

### cc create inventory components attributes

Adds a new search attribute for a specified run-time component. The command supports only single-valued search attributes. A run-time component can have several search attributes, but each attribute takes a single value, for example:

```
attribute1=value1
attribute2=value2
attribute3=value3
...
```

#### Syntax

- Command Central syntax:

```
cc create inventory components attributes node_alias componentid [attribute=value]
[--input | -i] filename{.xml|.json} [options]
```

- Not supported on Platform Manager.

## Arguments and Options

Argument or Option	Description
<code>node_alias</code>	<p>Required. Specifies the alias name of the installation for which you want to add component information.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<code>componentid</code>	<p>Required. Specifies the ID of the run-time component for which you want to create a search attribute.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<code>[attribute=value]</code>	<p>Optional. The name of the new search attribute and its matching value.</p> <p>If the search attribute already exists, the command returns an error.</p>
<code>[{--input   -i} filename{.xml .json}]</code>	<p>Optional. Identifies an input file that contains the data for the new search attribute.</p> <p>You retrieve attribute data for the <code>.xml   json</code> file using the <code>cc get inventory components attributes</code> command:</p> <p>For more information, see <a href="#">"--input   -i" on page 70</a>.</p>
<code>[options]</code>	<p>Optional. The command supports all options supported by Command Central. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a>.</p>

## Usage Notes

After creating a new search attribute for a run-time component, you can include the new attribute as search criteria in a lifecycle or search command to execute an operation against all run-time components that match the new search attribute. For example, the `cc execute lifecycle start group=AB` command starts all run-time components included in the "AB" group.

For information about including search criteria in lifecycle commands, see ["Specifying Search Criteria for Lifecycle Commands" on page 239](#). For information about

including search criteria in search inventory commands, see ["Specifying Search Criteria for Inventory Commands" on page 201](#).

### Examples When Executing on Command Central

- To create two search attributes, one with name "group" that matches the value "Test" and another with name "tenant" that matches the value "abc.com", for the run-time component that has the component ID "OSGI-SPM" and is installed on the installation with the alias name "sag01":

```
cc create inventory components attributes
sag01 OSGI-SPM group=Test tenantId=abc.com --password secret
```

- To create a new search attribute, using the attribute data from the attributes.xml file, for the run-time component that has the component ID "OSGI-CCE" and is installed on the installation with the alias name "sag01":

```
cc create inventory components attributes
sag01 OSGI-CCE -i c:\inputs\attributes.xml --password secret
```

To retrieve attribute data for the attribute.xml file use the following command:

```
cc get inventory components attributes sag01 OSGI-CCE -f xml
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see ["--username | -u" on page 80](#). The command specifies "secret" for the user's password.

### Related Commands

[cc list inventory components attributes](#)

[cc update inventory components attributes](#)

[cc delete inventory components attributes](#)

## cc get inventory components

Retrieves information about a specified run-time component. Information about a run-time component can include:

- Display name
- ID for the run-time component
- ID of the product to which this run-time component belongs
- Run-time component category, which can be one of the following:
  - `PROCESS` for run-time components that functions on its own. These are referred to as *instances* in the Web user interface.
  - `ENGINE` for run-time components that cannot function on their own, but rather run within a `PROCESS` run-time component. These are referred to as *components* in the Web user interface.

## Syntax

### ■ Command Central syntax:

```
cc get inventory components node_alias componentid [options]
```

### ■ Platform Manager syntax:

```
cc get inventory components componentid [options]
```

```
options :
[--accept | -a] content_type]
[--debug | -d]
[--error | -r] file]
[--format | -f] {tsv args | xml | csv args | json}]
[--log | -l] file]
[--output | -o] file]
[--password | -p] password]
[--quiet | -q]
[--server | -s] url]
[--username | -u] user_name]
```

## Arguments and Options

Argument or Option	Description
<i>node_alias</i>	<p>Command Central only.</p> <p>Required. Specifies the alias name of the installation for which you want to retrieve component information.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<i>componentid</i>	<p>Required. Specifies the ID of the run-time component for which you want to retrieve information.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
[ <i>options</i> ]	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands"</a> on page 59.</p>

## Example When Executing on Command Central

To execute a command on the Command Central server with host name "rubicon" and port "8090" to retrieve information for the run-time component that has the component ID "OSGI-SPM" and is installed on the installation with the alias name "sag01", and have the output returned to the console in JavaScript Object Notation format:

```
cc get inventory components sag01 OSGI-SPM --server http://rubicon:8090/cee
--format json --password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see ["--username | -u" on page 80](#). The command specifies "secret" for the user's password.

### Example When Executing on Platform Manager

To retrieve information for the run-time component that has the component ID "OSGI-SPM" and is managed by the Platform Manager with host name "rubicon2" and port "8092", using the authorization of the user with user name "Administrator" and password "manage", and have the information displayed on the console in XML format:

```
cc get inventory components OSGI-SPM --server http://rubicon2:8092/spm
--format xml --username Administrator --password manage
```

### Related Commands

[cc list inventory components](#)

[cc list instances](#)

[cc update inventory components](#)

[cc update instances](#)

## cc get inventory components attributes

Retrieves the attribute value matching a search attribute for a run-time component. To retrieve data for all search attributes for a run-time component, use the [cc list inventory components attributes](#) command.

### Syntax

- Command Central syntax:

```
cc get inventory components attributes node_alias componentid
[attribute] [options]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	<p>Required. Specifies the alias name of the installation for which you want to retrieve component information.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>

Argument or Option	Description
<i>componentid</i>	<p>Required. Specifies the ID of the run-time component for which you want to retrieve search attribute data.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
[ <i>attribute</i> ]	<p>Optional. The name of the search attribute whose matching value you want to retrieve.</p> <p>When you do not specify a value for the attribute argument, the command lists all attributes for a run-time component and their matching values.</p>
[ <i>options</i> ]	<p>Optional. The command supports all options supported by Command Central. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a>.</p>

### Example When Executing on Command Central

- To retrieve the value for a search attribute with name “group” for the run-time component that has component ID “OSGI-SPM” and is installed on the installation with the alias name “sag01”:

- To include the name “group” and value “AB” of the search attribute with headers in the output:

```
cc get inventory components attributes sag01 OSGI-SPM
group -p secret
```

#### Output:

```
Name      Value
group     AB
```

- To include only the value of the “group” attribute, for example “AB”, without any headers in the output:

```
cc get inventory components attributes sag01 OSGI-SPM
group properties=value includeHeaders=false -p secret
```

- To retrieve all search attributes and their matching values for the run-time component that has component ID “OSGI-SPM” and is installed on the installation with the alias name “sag01”, and has two search attributes “group” and “tenantId”:

```
cc get inventory components attributes sag01 OSGI-SPM -p manage
```

#### Output:

```
Name      Value
group     AB
tenantId  1234
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.

### Related Commands

[cc create inventory components attributes](#)

[cc list inventory components attributes](#)

[cc list instances](#)

[cc update inventory components attributes](#)

## cc list inventory components

Lists information about run-time components. Information about a run-time component can include:

- Display name
- ID for the run-time component
- ID of the product to which this run-time component belongs
- Run-time component category, which can be one of the following:
  - PROCESS for run-time components that functions on its own. These are referred to as *instances* in the Web user interface.
  - ENGINE for run-time components that cannot function on their own, but rather run within a PROCESS run-time component. These are referred to as *components* in the Web user interface.

### Syntax

- Command Central syntax:

- To list components for a specified installation:

```
cc list inventory components [node_alias] [componentid]
[options]
```

- To list components that match specified search criteria:

```
cc list inventory components [criteria] [start=number]
[size=number] [options]
```

- Platform Manager syntax:

```
cc list inventory components [componentid] [options]
```

```
options:
[{-#accept | -a} content_type]
[{-#debug | -d}]
[{-#error | -r} file]
[{-#format | -f} {tsv args | xml | csv args | json}]
[{-#log | -l} file]
[{-#output | -o} file]
```

```
[{--password | -p} password]
[{--quiet | -q}]
[{--server | -s} url]
[{--username | -u} user_name]
```

## Arguments and Options

Argument or Option	Description
[ <i>node_alias</i> ]	<p>Command Central only.</p> <p>Optional. Specifies the alias name of the installation for which you want to retrieve component information.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p> <p>If you do not specify an alias name nor search criteria, the command lists information for all run-time components for all installations that Command Central manages.</p>
[ <i>componentid</i> ]	<p>Optional. Specifies the ID of the run-time component for which you want to retrieve information.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
[ <i>criteria</i> ]	<p>Command Central only.</p> <p>Optional. Narrows down the list of returned run-time components to only those that match the search criteria you specify. For more information, see "<a href="#">Specifying Search Criteria for Inventory Commands</a>" on page 201.</p>
[ <i>start=number</i> ]	<p>Command Central only.</p> <p>Optional. Limits the results the command returns those starting at specified number in the results.</p> <p>For example, if you want to return information for the 5th through the 8th run-time components in the results, use <code>start=5 size=4</code>.</p>
[ <i>size=number</i> ]	<p>Command Central only.</p> <p>Optional. Limits the number of results you want returned.</p>

Argument or Option	Description
	For example, if you specify <code>size=1</code> , the command returns information for only one run-time component.
[options]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Examples When Executing on Command Central

- To list all run-time components that the Command Central with host name “rubicon” and port “8090” manages, using the authorization of the user with user name “Administrator” and password “manage”, and have the information returned to the output file “components” in XML format:

```
cc list inventory components --format xml --output components
--server http://rubicon:8090/cce --username Administrator
--password manage
```

- To list the same run-time components as the first example above, but restrict the number of returned run-time components to only 5:

```
cc list inventory components size=5 --format xml --output components
--server http://rubicon:8090/cce --username Administrator
--password manage
```

- To list the 10th through the 15th run-time components in the results and return the output to the console in XML format:

```
cc list inventory components start=10 size=6 --format xml
--password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see ["--server | -s" on page 77](#) and ["--username | -u" on page 80](#). The command specifies “secret” for the user’s password.

- To list run-time components and use search criteria to narrow the results to only those that are installed in the installation with alias name “sag01” and that have the component ID “OSGI-CCE” and return the output to the console in JavaScript Object Notation format:

```
cc list inventory components nodeAlias=sag01
runtimeComponentId=OSGI-CCE --format json --password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see ["--server | -s" on page 77](#) and ["--username | -u" on page 80](#). The command specifies “secret” for the user’s password.

- To list run-time components and use search criteria to narrow the results to only those that are installed in the installation with alias names “sag01” or “sag03” and return the output to the console in xml format:

```
cc list inventory components logicalOperator=OR nodeAlias=sag01
nodeAlias=sag03 --format xml --password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see “[--server | -s](#)” on page 77 and “[--username | -u](#)” on page 80. The command specifies “secret” for the user’s password.

### Example When Executing on Platform Manager

To list all run-time components managed by the Platform Manager with host name “rubicon2” and port “8092”, using the authorization of the user with user name “Administrator” and password “manage”, and have the information returned to the console in XML format:

```
cc list inventory components --format xml
--server http://rubicon2:8092/spm --username Administrator
--password manage
```

### Related Commands

[cc get inventory components](#)

[cc list instances](#)

[cc update inventory components](#)

[cc update instances](#)

## Specifying Search Criteria for Inventory Commands

When using the `cc list inventory components` or `cc list inventory products` commands to list run-time components or products, you can specify search criteria to narrow the results that the command returns. Supply the search criteria using the following format:

```
attribute1=value1 attribute2=value2 ...
```

For the search criteria, you specify attribute values to match, for example `runtimeComponentId=OSGI-CCE`, where `runtimeComponentId` is the attribute and the value to match is `OSGI-CCE`.

### Attribute Names You Can Use in the Search Criteria

Command	Attribute Names
<code>cc list inventory components</code>	<ul style="list-style-type: none"> <li>■ <code>nodeName</code></li> <li>■ <code>nodeAlias</code></li> <li>■ <code>nodeUrl</code></li> <li>■ <code>environmentName</code></li> </ul>

Command	Attribute Names
	<ul style="list-style-type: none"> <li>■ environmentAlias</li> <li>■ runtimeComponentInfoId</li> <li>■ runtimeComponentId</li> <li>■ runtimeComponentDisplayName</li> <li>■ runtimeComponentProductId</li> <li>■ runtimeComponentCategory</li> <li>■ runtimeComponentRuntimeStatus</li> <li>■ runtimeComponentRuntimeParentId</li> </ul> <p>You can combine any of the pre-defined attribute names in this list with new search attributes added using the <code>cc create inventory components attributes</code> command. For more information about syntax and usage, see <a href="#">cc create inventory components attributes</a>.</p>
<code>cc list inventory fixes</code>	<ul style="list-style-type: none"> <li>■ nodeName</li> <li>■ nodeAlias</li> <li>■ nodeUrl</li> <li>■ environmentName</li> <li>■ environmentAlias</li> <li>■ fixId</li> <li>■ fixDisplayName</li> <li>■ fixVersion</li> <li>■ fixGroup</li> <li>■ fixProducts</li> </ul>
<code>cc list inventory products</code>	<ul style="list-style-type: none"> <li>■ nodeName</li> <li>■ nodeAlias</li> <li>■ nodeUrl</li> <li>■ environmentName</li> <li>■ environmentAlias</li> <li>■ productId</li> <li>■ productCanonicalId</li> <li>■ productDisplayName</li> <li>■ productParentId</li> <li>■ productGroup</li> <li>■ productProfileDir</li> <li>■ productCode</li> <li>■ productVersion</li> <li>■ productInstallTime</li> </ul>

### Specifying the Value

When specifying the value, you can include the `*` pattern-matching character to match multiple characters or the `?` character to match a single character. For example, if you

want to narrow the list of returned products to only those with “mws” anywhere in their product display names, use the following search criterion:

```
productDisplayName=*mws*
```

**Important:** The search is case-sensitive.

You can also use the \* and ? pattern-matching characters to search for attribute names. For example, if you want to list the search attributes `attribute1=value1` and `attribute2=value2` for different run-time components, use the following search criterion:

```
attribute?=value?
```

### Logical Operators Used When Specifying Multiple Search Properties

If you specify multiple search items, by default, the command performs an AND operation to return results that match all the specified criteria. For example, to narrow the list of returned products to those with “mws” anywhere in their product display names and that are version 9.0 or later, use the following search criteria:

```
productDisplayName=*mws* productVersion=9.0*
```

You can use an OR operation with two attributes. To do so, specify the `logicalOperator=OR` argument. For example, to narrow the list of returned run-time components to those installed in installations that have the alias name “sag01” or “sag02”, use the following search criteria:

```
nodeAlias=sag01 logicalOperator=OR nodeAlias=sag02
```

### Related Commands

[cc list inventory components](#)

[cc list inventory products](#)

[cc list inventory fixes](#)

## cc list inventory components attributes

Lists all search attributes for a run-time component.

### Syntax

- Command Central syntax:

```
cc list inventory components attributes node_alias componentid [options]
```

- Not supported on Platform Manager.

## Arguments and Options

Argument or Option	Description
<code>node_alias</code>	<p>Required. Specifies the alias name of the installation for which you want to list component information.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<code>componentid</code>	<p>Required. Specifies the ID of the run-time component for which you want to list the existing search attributes.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<code>[options]</code>	<p>Optional. The command supports all options supported by Command Central. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a>.</p>

## Examples When Executing on Platform Manager

- To list all search attributes for the run-time component that has the component ID "OSGI-SPM" and is installed on the installation with the alias name "sag01":

```
cc list inventory components attributes sag01 OSGI-SPM --password secret
```

- To list all search attributes for the run-time component that has the component ID "OSGI-CCE" and is installed on the installation with the alias name "sag01":

```
cc list inventory components attributes sag01 OSGI-CCE
-f xml -o attributes.xml --password secret
```

The command writes the output to the attributes.xml file. You can create or update search attributes for a run-time component in the attribute.xml file using the create or update inventory attributes commands, For example, to update the attributes in the attributes.xml file:

```
cc update inventory components attributes node_alias componentid -i attributes.xml
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see ["--username | -u" on page 80](#). The command specifies "secret" for the user's password.

## Related Commands

[cc create inventory components attributes](#)

[cc get inventory components attributes](#)

## cc update inventory components attributes

### cc update inventory components

Updates the display name and/or icon associated with a specified run-time component.

#### Syntax

- Command Central syntax:

```
cc update inventory components node_alias componentid
  [--input | -i] filename{.xml|.json|.properties} [options]

  options :
  [--debug | -d]
  [--error | -r] file
  [--log | -l] file
  [--password | -p] password
  [--quiet | -q]
  [--server | -s] url
  [--username | -u] user_name
```

- Not supported on Platform Manager.

#### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	<p>Required. Specifies the alias name of the installation for which you want to retrieve component information.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<i>componentid</i>	<p>Required. Specifies the ID of the run-time component that you want to update.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<pre>--input   -i} <i>filename</i>{.xml  .json .properties}</pre>	<p>Required. Identifies an input file that contains the updated data for the run-time component. For more information, see "<a href="#">--input   -i</a>" on page 70.</p>
[ <i>options</i> ]	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see "<a href="#">Options for the Commands</a>" on page 59.</p>

## Usage Notes

- The information that you can update for a run-time component is the display name and icon.
- To update the icon for a run-time component, you supply the icon ID of the new icon. To determine the icon IDs of installed icons, use the [cc list resources icons](#) command.

## Example When Executing on Command Central

To update the run-time component with ID OSGI-SPM that is installed in the installation with alias name "sag01" using the data in the c:\inputs\component\_data.xml file:

```
cc update inventory components sag01 OSGI-SPM
--input c:\inputs\component_data.xml
```

Because the `{--server | -s}`, `{--username | -u}`, and `{--password | -p}` options are not specified, the command uses the default server, user name, and password. For more information, see "[--server | -s](#)" on page 77, "[--username | -u](#)" on page 80, and "[--password | -p](#)" on page 76.

## Related Commands

[cc get inventory components](#)

[cc list inventory components](#)

[cc list instances](#)

[cc list resources icons](#)

[cc update instances](#)

## cc update inventory components attributes

Updates the value that matches an existing search attribute for a run-time component.

### Syntax

- Command Central syntax:

```
cc update inventory components attributes node_alias componentid
[attribute=value] [--input | -i] filename{.xml|.json} [options]
```

- Not supported by Platform Manager.

## Arguments and Options

Argument or Option	Description
<code>node_alias</code>	<p>Required. Specifies the alias name of the installation for which you want to update component information.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<code>componentid</code>	<p>Required. Specifies the ID of the run-time component for which you want to update the value of a search attribute.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<code>[attribute=value]</code>	<p>Optional. The name of the search attribute and the new matching value for the property.</p>
<code>[{--input   -i} filename{.xml .json}]</code>	<p>Optional. Identifies an input file that contains the update data for the search attribute.</p> <p>You retrieve attribute data for the <code>.xml   json</code> file using the <code>c cc get inventory components attributes</code> command.</p> <p>For more information, see <a href="#">"--input   -i" on page 70</a>.</p>
<code>[options]</code>	<p>Optional. The command supports all options supported by Command Central. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a>.</p>

## Usage Notes

You can execute the command either with the `[attribute=value]`, or the `[{--input | -i} filename{.xml|.json}]` argument, but not with both arguments in the same command.

## Examples When Executing on Command Central

- To change the existing value of a search property with name "group" to the new value "Production" for the run-time component that has the component ID "OSGI-SPM" and is installed on the installation with the alias name "sag01":

```
cc update inventory components attributes sag01 OSGI-SPM
```

```
group=Production --password secret
```

- To update the value of a search property, using the property data from the `attributes.xml` file, for the run-time component that has the component ID “OSGI-CCE” and is installed on the installation with the alias name “sag01”:

```
cc update inventory components attributes sag01 OSGI-CCE
-i c:\inputs\attributes.xml --password secret
```

- To update a search property with name “group”, using the property data from the `attributes.xml` file, for the run-time component that has the component ID “OSGI-SPM” and is installed on the installation with the alias name “sag01”:

```
cc update inventory components attributes sag01 OSGI-SPM
group -i c:\inputs\attributes.xml --password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see ["--username | -u" on page 80](#). The command specifies “secret” for the user’s password.

### Related Commands

[cc create inventory components attributes](#)

[cc get inventory components attributes](#)

[cc list inventory components attributes](#)

[cc delete inventory components attributes](#)

## cc delete inventory components attributes

Deletes an existing search attribute for a run-time component.

### Syntax

- Command Central syntax:

```
cc delete inventory components attributes node_alias componentid
[attribute] [options]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
<code>node_alias</code>	<p>Required. Specifies the alias name of the installation for which you want to delete component information.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>

Argument or Option	Description
<i>componentid</i>	Required. Specifies the ID of the run-time component for which you want to delete a search attribute.  You can determine the IDs for run-time components using <a href="#">cc list inventory components</a> .
[attribute]	Optional. The name of the search attribute that you want to delete.
[options]	Optional. The command supports all options supported by Command Central. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Usage Notes

If you do not specify a search attribute name, the command deletes all existing search attributes for a run-time component.

### Examples When Executing on Command Central

- To delete a search attribute with name "group" for the run-time component that has the component ID "OSGI-SPM" and is installed on the installation with the alias name "sag01":

```
cc delete inventory components attributes sag01
OSGI-SPM group --password secret
```

- To delete all search attributes for the run-time component that has the component ID "OSGI-CCE" and is installed on the installation with the alias name "sag01":

```
cc delete inventory components attributes sag01
OSGI-CCE --password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.

### Related Commands

[cc get inventory components attributes](#)

[cc list inventory components attributes](#)

[cc update inventory components attributes](#)

## cc get inventory fixes compare

Compares the fixes installed in two or more installations.

### Syntax

#### ■ Command Central syntax:

```
cc get inventory fixes compare nodeAlias=alias1 nodeAlias=alias2
[nodeAlias=alias3 ...nodeAlias=aliasn] [options]
```

```
options:
[--accept | -a] content_type
[--debug | -d]
[--error | -r] file
[--format | -f] {tsv args | xml | csv args | json}
[--log | -l] file
[--output | -o] file
[--password | -p] password
[--quiet | -q]
[--server | -s] url]
[--username | -u] user_name
```

#### ■ Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
nodeAlias= <i>alias1</i> nodeAlias= <i>alias2</i> [nodeAlias= <i>alias3</i> ... nodeAlias= <i>aliasn</i> ]	Required. Specifies the alias names of two or more installations that you want to compare.  You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a> .
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands"</a> on page 59.

### Usage Notes

- The command returns the results of the comparison.

### Example When Executing on Command Central

To execute a command on the Command Central server with host name "rubicon" and port "8090" to compare the fixes applied to the installations with alias names "sag01" and "sag02" and have the output returned to the console in XML format:

```
cc get inventory fixes compare nodeAlias=sag01 nodeAlias=sag02
--server http://rubicon:8090/cce --format xml --password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.

## Related Commands

[cc list inventory fixes](#)

## cc list inventory fixes

Lists information about fixes that have been applied to products. Information about fixes can include:

- Fix ID
- Fix name
- Version of the fix
- Product to which the fix is applied

## Syntax

- Command Central syntax:

- To list information for a specified fix:

```
cc list inventory fixes [fixid] [options]
```

- To list fixes that match specified search criteria:

```
cc list inventory fixes [criteria] [start=number] [size=number]
[options]
```

- Platform Manager syntax:

```
cc list inventory fixes [fixid] [options]
```

```
options:
[{--accept | -a} content_type]
[{--debug | -d}]
[{--error | -r} file]
[{--format | -f} {tsv args | xml | csv args | json}]
[{--log | -l} file]
[{--output | -o} file]
[{--quiet | -q}]
[{--password | -p} password]
[{--server | -s} url]
[{--username | -u} user_name]
```

## Arguments and Options

Argument or Option	Description
<i>fixid</i>	Optional. Specifies the ID of the fix for which you want information.

Argument or Option	Description
<code>[criteria]</code>	<p>Command Central only.</p> <p>Optional. Narrows down the list of returned fixes to only those that match the search criteria you specify. For more information, see <a href="#">"Specifying Search Criteria for Inventory Commands"</a> on page 201.</p>
<code>[start=number]</code>	<p>Command Central only.</p> <p>Optional. Limits the results the command returns those starting at specified number in the results.</p> <p>For example, if you want to return information for the 5th through the 8th products in the results, use <code>start=5 size=4</code>.</p>
<code>[size=number]</code>	<p>Command Central only.</p> <p>Optional. Limits the number of results you want returned.</p> <p>For example, if you specify <code>size=1</code>, the command returns information for only one product.</p>
<code>[options]</code>	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands"</a> on page 59.</p>

### Examples When Executing on Command Central

- To execute a command on the Command Central server with host name "rubicon" and port "8090" to retrieve information for the fix with ID "CCE\_9-0\_Fix2" and have the output returned to the console in JavaScript Object Notation format:

```
cc get inventory fixes CCE_9-0_Fix2 --server http://rubicon:8090/cce
--format json --password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see ["--username | -u" on page 80](#). The command specifies "secret" for the user's password.

- To list the fixes applied to all products that the Command Central with host name "rubicon" and port "8090" manages, using the authorization of the user with user name "Administrator" and password "manage", and have the information returned to the output file "fixlist" in XML format:

```
cc list inventory fixes --format xml --output fixlist
--server http://rubicon:8090/cce --username Administrator
--password manage
```

- To list the same fixes as the first example above, but restrict the number of returned products to only 5:

```
cc list inventory fixes size=5 --format xml --output productlist
--server http://rubicon:8090/cce --username Administrator
--password manage
```

- To list the fixes that are version 9.0 or later and also contain “wMFix” in their fix IDs, and return the output to the console in JavaScript Object Notation format:

```
cc list inventory fixes fixversion=9.0* fixId=*wMFix* --format json
--password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see ["--server | -s" on page 77](#) and ["--username | -u" on page 80](#). The command specifies “secret” for the user’s password.

- To list the 10th through the 15th fixes in the results and return the output to the console in XML format:

```
cc list inventory fixes start=10 size=6 --format xml --password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see ["--server | -s" on page 77](#) and ["--username | -u" on page 80](#). The command specifies “secret” for the user’s password.

### Example When Executing on Platform Manager

To list information about the fixes applied to all the products that are managed by the Platform Manager with host name “rubicon2” and port “8092”, using the authorization of the user with user name “Administrator” and password “manage”, and have the information returned to the console in XML format:

```
cc list inventory fixes --format xml --username Administrator
--password manage --server http://rubicon2:8092/spm
```

### Related Commands

[cc get inventory fixes compare](#)

## cc get inventory products

Retrieves information about a specified product. Information about a product can include:

- Display name
- ID for the product
- Product code
- Product version
- Date and time the product was installed

## Syntax

### ■ Command Central syntax:

```
cc get inventory products node_alias productid [options]
```

### ■ Platform Manager syntax:

```
cc get inventory products productid [options]
```

```
options :
[{--accept | -a} content_type]
[{--debug | -d}]
[{--error | -r} file]
[{--format | -f} {tsv args | xml | csv args | json}]
[{--log | -l} file]
[{--output | -o} file]
[{--password | -p} password]
[{--quiet | -q}]
[{--server | -s} url]
[{--username | -u} user_name]
```

## Arguments and Options

Argument or Option	Description
<i>[node_alias]</i>	<p>Command Central only.</p> <p>Required. Specifies the alias name of the installation for which you want to retrieve product information. If you do not specify an alias name, the command lists all products in all installations that Command Central manages.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<i>productid</i>	<p>Required. Specifies the product ID of the product for which you want to retrieve product information.</p> <p>You can determine the IDs for products using <a href="#">cc list inventory products</a>.</p>
<i>[options]</i>	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a>.</p>

## Example When Executing on Command Central

To execute a command on the Command Central server with host name “rubicon” and port “8090” to retrieve information for the product that has the ID “SPM” and is

installed on the installation with the alias name “sag01”, and have the output returned to the console in JavaScript Object Notation format:

```
cc get inventory products sag01 SPM --server http://rubicon:8090/cce
--format json --password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see “[--username | -u](#)” on page 80. The command specifies “secret” for the user’s password.

### Example When Executing on Platform Manager

To retrieve information for the product that has the ID “SPM” and is managed by the Platform Manager with host name “rubicon2” and port “8092”, and have the output returned to the console in XML format:

```
cc get inventory products SPM --server http://rubicon2:8092/spm
--format xml --password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see “[--username | -u](#)” on page 80. The command specifies “secret” for the user’s password.

### Related Commands

[cc get inventory products compare](#)

[cc list inventory products](#)

## cc get inventory products compare

Compares the products installed in two or more installations.

### Syntax

- Command Central syntax:

```
cc get inventory products compare nodeAlias=alias1 nodeAlias=alias2
[nodeAlias=alias3 ... nodeAlias=aliasn] [options]
```

```
options:
[{--accept | -a} content_type]
[{--debug | -d}]
[{--error | -r} file]
[{--format | -f} {tsv args | xml | csv args | json}]
[{--log | -l} file]
[{--output | -o} file]
[{--password | -p} password]
[{--quiet | -q}]
[{--server | -s} url]]
[{--username | -u} user_name]
```

- Not supported by Platform Manager.

## Arguments and Options

Argument or Option	Description
<pre>nodeAlias=<i>alias1</i> nodeAlias=<i>alias2</i> [<i>nodeAlias=alias3</i> ... nodeAlias=<i>aliasn</i>]</pre>	<p>Required. Specifies the alias names of two or more installations that you want to compare.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<pre>[<i>options</i>]</pre>	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands"</a> on page 59.</p>

## Usage Notes

- The command returns the results of the comparison.

## Example When Executing on Command Central

To execute a command on the Command Central server with host name “rubicon” and port “8090” to compare the products installed in the installations with alias names “sag01” and “sag02” and have the output returned to the console in XML format:

```
cc get inventory products compare nodeAlias=sag01 nodeAlias=sag02
--server http://rubicon:8090/cce --format xml --password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see ["--username | -u"](#) on page 80. The command specifies “secret” for the user’s password.

## Related Commands

[cc get inventory products](#)

[cc list inventory products](#)

## cc list inventory products

Lists information about products. Information about a product can include:

- Display name
- ID for the product
- Product code
- Product version
- Date and time the product was installed

## Syntax

### ■ Command Central syntax:

#### ■ To list products for a specified installation:

```
cc list inventory products [node_alias] [productid] [options]
```

#### ■ To list products that match specified search criteria:

```
cc list inventory products [criteria] [start=number]
[size=number] [options]
```

### ■ Platform Manager syntax:

```
cc list inventory products [productid] [options]
```

```
options:
[--accept | -a] content_type]
[--debug | -d]
[--error | -r] file]
[--format | -f] {tsv args | xml | csv args | json}]
[--log | -l] file]
[--output | -o] file]
[--password | -p] password]
[--quiet | -q]
[--server | -s] url]
[--username | -u] user_name]
```

## Arguments and Options

Argument or Option	Description
[ <i>node_alias</i> ]	<p>Command Central only.</p> <p>Optional. Specifies the alias name of the installation for which you want to retrieve product information.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p> <p>If you do not specify an alias name nor search criteria, the command lists all products in all installations that Command Central manages.</p>
[ <i>productid</i> ]	<p>Optional. Specifies the ID of the product for which you want to retrieve information.</p>
[ <i>criteria</i> ]	<p>Command Central only.</p> <p>Optional. Narrows down the list of returned products to only those that match the search criteria you specify. For more information,</p>

Argument or Option	Description
	see <a href="#">"Specifying Search Criteria for Inventory Commands"</a> on page 201.
[start= <i>number</i> ]	<p>Command Central only.</p> <p>Optional. Limits the results the command returns those starting at specified number in the results.</p> <p>For example, if you want to return information for the 5th through the 8th products in the results, use <code>start=5 size=4</code>.</p>
[size= <i>number</i> ]	<p>Command Central only.</p> <p>Optional. Limits the number of results you want returned.</p> <p>For example, if you specify <code>size=1</code>, the command returns information for only one product.</p>
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands"</a> on page 59.

### Examples When Executing on Command Central

- To list all products that the Command Central with host name "rubicon" and port "8090" manages, using the authorization of the user with user name "Administrator" and password "manage", and have the information returned to the output file "productlist" in XML format:

```
cc list inventory products --format xml --output productlist
--server http://rubicon:8090/cce --username Administrator
--password manage
```

- To list the same products as the first example above, but restrict the number of returned products to only 5:

```
cc list inventory products size=5 --format xml --output productlist
--server http://rubicon:8090/cce --username Administrator
--password manage
```

- To list products that are version 9.0 or later and also contain "Platform" in their display name and return the output to the console in JavaScript Object Notation format:

```
cc list inventory products productVersion=9.0*
productDisplayName=*Platform* --format json --password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see "[--server | -s](#)" on page 77 and "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.

- To list the 10th through the 15th products in the results and return the output to the console in XML format:

```
cc list inventory products start=10 size=6 --format xml
--password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see "[--server | -s](#)" on page 77 and "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.

### Example When Executing on Platform Manager

To list information about the products that are managed by the Platform Manager with host name "rubicon2" and port "8092", using the authorization of the user with user name "Administrator" and password "manage", and have the information returned to the console in XML format:

```
cc list inventory products --format xml --username Administrator
--password manage --server http://rubicon2:8092/spm
```

### Related Commands

[cc get inventory products compare](#)

[cc get inventory products](#)

## Resources Commands

### cc list resources icons

Lists information about the installed icons. This command is useful if you want to use the [cc update inventory components](#) command to update the icon associated with a run-time component and you need to determine an icon ID.

#### Syntax

- Command Central syntax:

```
cc list resources icons [options]

options:
[{--debug | -d}]
[{--error | -r} file]
[{--format | -f} {tsv args | text | xml | csv args | json}]
[{--log | -l} file]
[{--output | -o} file]
[{--password | -p} password]
[{--quiet | -q}]
```

```
[{--server | -s} url ]  
[{--username | -u} user_name ]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Example When Executing on Command Central

To list the icons available for the run-time components managed by the Command Central with host name "rubicon" with port "8090" and have the output returned to the console in XML format:

```
cc list resources icons --server http://rubicon:8090/cce --format xml  
--password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see ["--username | -u" on page 80](#). The command specifies "secret" for the user's password.

### Related Commands

[cc update inventory components](#)

---

# 8 Monitoring Instances

---

■ About Monitoring Instances .....	222
■ Viewing the Status of an Instance or Its Components .....	222
■ Modifying the Status of an Instance or Its Components .....	222
■ About Monitoring KPIs .....	223
■ Viewing Alerts for an Instance or Its Components .....	223
■ Clearing Alerts for an Instance or Its Components .....	225
■ About Monitoring-Related Events .....	225
■ Monitoring KPIs of Platform Manager Instances .....	226
■ Monitoring Commands .....	227

## About Monitoring Instances

---

Command Central allows you to monitor the overall health of an instance or its components. You can view and modify the statuses and alerts, and you can view the KPIs (key performance indicators) of an instance or its components. The status, alert, and KPI information is normally retrieved regularly by a polling mechanism from the instance or its component, but Command Central also reacts to monitoring-related events.

## Viewing the Status of an Instance or Its Components

---

An instance or instance component can have one of the following statuses:

- **Online:** The instance or instance component is currently running and the ping operation succeeds.
- **Failed:** The instance or instance component is not running and the ping operation fails.
- **Starting:** The instance or instance component is starting.
- **Stopped:** The instance or instance component has stopped.
- **Stopping:** The instance or instance component is stopping.
- **Unknown:** The status of the instance or instance component cannot be determined.
- **Unresponsive:** The ping operation fails, but other indicators such as the process-id file indicate that the instance or instance component is running.

---

### To view the status of an instance or its components

1. In the **Environments** pane, select the environment that contains the instance or instance component that you want to monitor.
2. Click the **Instances** tab.

The **Status** field in the table shows the status of the instance. To see the status of its components, expand the instance node.

## Modifying the Status of an Instance or Its Components

---

You can modify the status of an instance or its component. For more information, see "[Starting, Stopping, Pausing, Resuming, and Debugging Instances](#)" on page 234.

---

## About Monitoring KPIs

---

Command Central allows you to view up to three basic KPIs (key performance indicators) for each instance or instance component that is in online status.

Each KPI consists of the following information:

- Name
- Current value
- Marginal threshold
- Critical threshold
- Maximum value

KPIs are displayed as bar charts. A bar can have one of the following colors:

- **Green:** The current value is below the marginal threshold, indicating normal operation.
- **Yellow:** The current value is above the marginal threshold, indicating that performance or stability may be affected if it rises further.
- **Red:** The current value is above the critical threshold, indicating that performance or stability are probably impacted as a result.

KPIs are provided for the following products:

- **Software AG Platform Manager** For more information, see "[Monitoring KPIs of Platform Manager Instances](#)" on page 226.
- **Broker Server** For more information, see "[Monitoring webMethods Broker KPIs](#)" on page 401.
- **Integration Server** For more information, see "[Monitoring KPIs of Integration Server Instances](#)" on page 446.
- **My webMethods Server** For more information, see "[Monitoring KPIs of My webMethods Server Instances](#)" on page 463.

---

## Viewing Alerts for an Instance or Its Components

---

Command Central indicates whether there is an alert for an instance or for one of its components.

Alerts are raised or disabled when the status of an instance or instance component changes. In this case, the alert behavior is as follows:

oldStatus	newStatus	Alert	Severity
online	stopped	on	warning
online	unresponsive	on	error
online	failed	on	error
online	unknown	on	warning
not online	online	off	info

Alerts are also raised or disabled when the value of a KPI (key performance indicator) changes. In this case, the alert behavior is as follows:

oldZone	newZone	Alert	Severity
normal	marginal	on	warning
marginal	critical	on	error
normal	critical	on	error
critical	marginal	off	warning
marginal	normal	off	info
critical	normal	off	info

#### To view the alerts for an instance or its components

1. In the **Environments** pane, select the environment that contains the instance or instance component that you want to monitor.
2. Click the **Instances** tab.

**Note:** If there is an alert for an instance or instance component, the respective **Alert** field shows a flag.

3. To see more information about the alert, select an instance or an instance component from the table.
4. Click the **Overview** tab.

5. The **Alerts** field in the **Dashboard** shows the number of alerts. Point to the number to see the message texts and dates.

## Clearing Alerts for an Instance or Its Components

Command Central allows you to clear the alerts for an instance or for one of its components.

### To clear the alerts for an instance or its components

1. In the **Environments** pane, select the environment that contains the instance or instance component that you want to monitor.
2. Click the **Instances** tab.

**Note:** If there is an alert for an instance or instance component, the respective **Alert** field shows a flag.

3. Select an instance or an instance component from the table.
4. Click the **Overview** tab.
5. The **Alerts** field in the **Dashboard** shows the number of alerts. Click the number.
6. In the resulting pop-up window, click **Clear**.

## About Monitoring-Related Events

The monitoring information is normally retrieved by a polling mechanism, but Command Central also reacts to monitoring-related events. To use monitoring-related events, you must configure NERV (Network for Event Routing and Variation) for each instance of Command Central and of Software AG Platform Manager as described in *Implementing Event-Driven Architecture with webMethods Products, Configuring NERV*.

Using monitoring-related events enables integrated solutions with other Software AG products or third party products.

The following event types exist for integrated solutions:

- **RuntimeStatusChange** This event type is emitted when the status of an instance or instance component changes.
- **RuntimeStateChange** This event type is emitted when the value of a KPI (key performance indicator) changes.
- **Alert** This event type is emitted when changes occur in the instance's or instance component's status or state. For detailed information, see "[Viewing Alerts for an Instance or Its Components](#)" on page 223.

For more information about the detailed structure of monitoring-related events, go to `Software AG_directory\common\EventTypeStore\WebM\PlatformManagement`.

## Monitoring KPIs of Platform Manager Instances

Platform Manager returns the following three KPIs:

Name	Marginal Value	Critical Value	Maximum Value	Limitation Description
<b>Used machine physical memory (in MB)</b>	80%	95%	Physical memory of machine.	On hosts running a 32-bit JRE on a 32-bit operating system with more than 2GB RAM, this KPI shows a value that is too low. This is due to a known issue in the Java SE Runtime Environment.
<b>Used machine disk space (in MB)</b>	80%	95%	Physical disk space of machine.	
<b>CPU Utilization</b>	80%	90%	100%	<ul style="list-style-type: none"> <li>■ This KPI is <i>only</i> displayed, if you use Java 7.</li> <li>■ This KPI is <i>not</i> displayed when running on HP-UX.</li> </ul>

### To view the KPIs of Platform Manager instances

1. On the **Environments** pane, select the environment you want to monitor.
2. Click the **Instances** tab.
3. In the table, select the Platform Manager you want to monitor.
4. Click the **Overview** tab.

The **Monitoring** section in the **Dashboard** shows the KPIs of the Platform Manager instance.

## Monitoring Commands

### cc get monitoring

Retrieves the run-time statuses, run-time states, or states of run-time components.

- Run-time status indicates whether a run-time component is running or not. Examples of a run-time status are ONLINE or STOPPED.
- Run-time state indicates the health of a run-time component by providing (key performance indicators (KPIs) for the component. Each KPI provides information about the current use, marginal use, critical use, and maximum use. For example, a component might display a KPI for the amount of memory that would include the current memory use, when memory use is considered marginal, when memory use is considered critical, and the maximum memory use allowed.
- State provides both the run-time status and the run-time state.

For a list and description of run-time statuses and run-time states for a specific run-time component, see information in this reference for the product with which the run-time component is associated.

#### Syntax

- Command Central syntax:

```
cc get monitoring {runtimestatus | runtimestate | state} node_alias
componentid [options]
```

- Platform Manager syntax:

```
cc get monitoring {runtimestatus | runtimestate | state} componentid
[options]
```

```
options:
[--check-every | -c} seconds]
[--debug | -d]
[--error | -r} file]
[--expected-values | -e} values]
[--format | -f} {tsv args | xml | csv args | json}]
[--log | -l} file]
[--output | -o} file]
[--password | -p} password]
[--quiet | -q]
[--server | -s} url]
[--username | -u} user_name]
[--wait | -w} seconds]
```

## Arguments and Options

Argument or Option	Description
<code>{runtimestatus   runtimestate   state}</code>	Required. Specifies whether you want to retrieve run-time statuses, run-time states, or state.
<code>node_alias</code>	Command Central only.  Required. Specifies the alias name of the installation on which the run-time component is installed.  You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a> .
<code>componentid</code>	Required for <code>runtimestatus</code> and <code>runtimestate</code> . Not applicable for <code>state</code> .  Specifies the ID of the run-time component for which you want to retrieve information.  You can determine the IDs for run-time components using <a href="#">cc list inventory components</a> .
<code>[options]</code>	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

## Usage Notes

- The following are general descriptions of run-time statuses:
    - FAILED when a run-time component failed, for example, ended unexpectedly.
    - ONLINE when a run-time component is running.
    - PAUSED when a run-time component is paused.
    - STARTING when a run-time component is starting.
    - STOPPED when a run-time component is not running.
    - STOPPING when a run-time component is stopping.
    - UNKNOWN when the status cannot be determined.
    - UNRESPONSIVE when a run-time component is running, but is unresponsive.
- Note:** A specific run-time component might support only a subset of the statuses.

## Examples When Executing on Command Central

- To retrieve the run-time status of the run-time component that has the ID “OSGI-SPM” and is installed in the installation with alias name “sag01” and have the output returned to the console in XML format:

```
cc get monitoring runtimestatus sag01 OSGI-SPM --format xml
--password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see "[--server | -s](#)" on page 77 and "[--username | -u](#)" on page 80. The command specifies “secret” for the user’s password.

- To initiate the shutdown of the Integration Server with the component ID “OSGI-IS” running in the installation “sag01”, then execute the `cc get monitoring runtimestatus` command to wait 60 seconds for the command to complete and return the expected results “STOPPED”, checking for results every 5 seconds:

```
cc exec lifecycle stop sag01 OSGI-IS --password secret
```

```
cc get monitoring runtimestatus sag01 OSGI-IS --expected-values STOPPED
--wait 60 --check-every 5 --password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see "[--server | -s](#)" on page 77 and "[--username | -u](#)" on page 80. The command specifies “secret” for the user’s password.

## Example When Executing on Platform Manager

To retrieve the state of the run-time component that has the ID “OSGI-SPM” and is installed in the installation that the Platform Manager server with host name “rubicon2” and port “8092” manages, and have the output returned to the console in XML format:

```
cc get monitoring state OSGI-SPM --format xml
--server http://rubicon2:8092/spm --password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see "[--username | -u](#)" on page 80. The command specifies “secret” for the user’s password.

## Product-Specific Information

[Run-time Monitoring Statuses for IntegrationServer-instanceName](#)

[Run-time Monitoring States for IntegrationServer-instanceName](#)

[Monitoring Run-time Statuses for webMethods Broker](#)

[Monitoring Run-time States for webMethods Broker](#)

## Related Commands

[cc list inventory components](#)

## cc list monitoring alerts

Lists the alerts for a specified run-time component.

### Syntax

- Command Central syntax:

```
cc list monitoring alerts [node_alias [componentid]] [options]
```

```
options:
[--debug | -d]
[--error | -r] file]
[--format | -f] {tsv args | xml | csv args | json}]
[--log | -l] file]
[--output | -o] file]
[--password | -p] password]
[--quiet | -q]
[--server | -s] url]]
[--username | -u] user_name]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
[ <i>node_alias</i> ] [ <i>componentid</i> ]	Optional. Specifies the alias name of the installation and the ID of the run-time component for which you want to retrieve information. To specify <i>componentid</i> , you must specify <i>node_alias</i> .  You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a> . You can determine the IDs for run-time components using <a href="#">cc list inventory components</a> .
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Examples When Executing on Command Central

To execute a command on the Command Central server with host name "rubicon" and port "8090" to list the alerts for the run-time component that has the ID "OSGI-SPM" and is installed in the installation with alias name "sag01", and have the output returned to the console in XML format:

```
cc list monitoring alerts sag01 OSGI-SPM --format xml
--server http://rubicon:8090/cce --password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.



---

# 9 Administering Product Lifecycle

---

■ About Administering Product Lifecycle .....	234
■ Starting, Stopping, Pausing, Resuming, and Debugging Instances .....	234
■ Viewing Product Logs .....	235
■ Lifecycle Commands .....	236

---

## About Administering Product Lifecycle

---

The following sections show how to centrally administer the lifecycle of managed products.

### Lifecycle Actions

The lifecycle actions are specific to a product instance. A lifecycle action is disabled if it is not applicable for a product instance. See the product-specific description for each of these actions.

Action	Description
<b>Start</b>	Starts an instance that was stopped or not started.
<b>Stop</b>	Stops an instance that was started earlier.
<b>Pause</b>	Pauses an instance that was started earlier.
<b>Restart</b>	Restarts an instance that was running or stopped earlier.
<b>Debug</b>	Starts stopped instance in debug mode.
<b>Resume</b>	Resumes an instance that was paused earlier. Resume works differently for different product instances. See the product-specific description for this action.
<b>Safe mode</b>	Runs an instance in safe mode for diagnostic purpose.

## Starting, Stopping, Pausing, Resuming, and Debugging Instances

---

Use the following procedure to change the status of an instance.

---

### To change the status of an instance

1. In the Environments pane, select the environment in which you want to change the status of an instance.
2. Select the **Instances** tab.

3. Expand the instance node and click the corresponding status icon.
4. Select the required action from the Lifecycle Actions dialog.  
Command Central performs the selected action on the instance through Platform Manager and lists the updated instance status on the **Instances** tab.  
You can also change the status of an instance in the **Overview** tab of the instance.

## Viewing Product Logs

You view run-time component or product logs with information about operations and errors that occur on the product using the Command Central web user interface or the Command Line tool.

### To view the available logs for a product

1. In the Environments pane, select the environment for the product you require.
2. On the **Instances** tab, click the name of the product instance.
3. On the product instance page, select the **Logs** tab.

Command Central lists the available product logs on the Log Sources page.

## Viewing the Contents of a Log

On the Log Sources page, you can view the contents of a log by clicking on the log alias in the Alias column.

In the search field, you can search for log entries by supplying either a regular expression or search text.

**Note:** When **Highlight** is selected, you cannot use regular expressions in searches.

By default, Command Central shows the last 100 log entries. Use the filtering options on the Logs page to view specific lines in the log as follows:

- When you select **Filter** and specify a regular expression or search text, Command Central shows the specified number of entries in the log that contain the search text. For example, if you type "JMX" in the search field and specify 20 lines, Command Central shows up to 20 log entries that contain the word "JMX".
- When you select **Filter**, but do not specify a regular expression or search text, Command Central shows the specified number of entries from the top or bottom of the log. For example, if the search field is empty and you select **First** and **20** lines, Command Central shows the first 20 entries in the log.
- When you select **Highlight** and specify a regular expression or search text, Command Central highlights the search text matches in the specified number of entries in the log that contain the search text.

**Note:** Switching from **Highlight** to **Filter** removes all highlighting from the text.

- When you select **Highlight**, but do not specify a regular expression or search text, Command Central shows the specified number of entries from the top or bottom of the log.

## Downloading a Log

You can download one or more logs at a time.

### To download a log

1. In the Download column on the Log Sources page, click  for the log that you want to download.
2. Select the format in which you want to download the log.

## Downloading Multiple Logs

### To download multiple logs

1. On the Log Sources page, select the logs you want to download.

**Note:** If no logs are selected, Command Central downloads all available logs.

2. Click .
3. Click **Download selected logs**.

## Lifecycle Commands

### cc exec lifecycle

Executes an action against run-time components. You can execute actions to start, stop, pause, and/or resume run-time components.

#### Syntax

- Command Central syntax:
  - To execute an action against a specified component:
 

```
cc exec lifecycle action node_alias componentid [options]
```
  - To execute an action against run-time components that meet specified search criteria:
 

```
cc exec lifecycle action [criteria] [options]
```
- Platform Manager syntax:

```

cc exec lifecycle components componentid action [options]

options:
[--accept | -a] content_type]
[--debug | -d]
[--error | -r] file]
[--format | -f] {tsv args | text | xml | csv args | json}]
[--log | -l] file]
[--output | -o] file]
[--password | -p] password]
[--quiet | -q]
[--server | -s] url]
[--username | -u] user_name]

```

## Arguments and Options

Argument or Option	Description
<i>action</i>	<p>Required. Specifies the action you want to take against the run-time component. Supply one of the following actions:</p> <ul style="list-style-type: none"> <li>■ <i>start</i> - starts the run-time component</li> <li>■ <i>startindebugmode</i> - starts the run-time component in debug mode</li> <li>■ <i>startinsafemode</i> - starts the run-time component in safe mode</li> <li>■ <i>stop</i> - stops the run-time component</li> <li>■ <i>restart</i> - stops, then restarts the run-time component</li> <li>■ <i>pause</i> - pauses the run-time component</li> <li>■ <i>resume</i> - resumes previously paused run-time component</li> </ul> <p>Run-time components might support all or just a subset of the actions. For information about the supported actions for a run-time component, see information in this reference for the product with which the run-time component is associated.</p>
<i>node_alias</i>	<p>Command Central only.</p> <p>Required when you do not specify search criteria. Specifies the alias name of an installation. You can determine installation alias names using the <a href="#">cc list landscape nodes</a> command.</p>
<i>componentid</i>	<p>Required. Specifies the component ID of a run-time component on which to act. You can determine the</p>

Argument or Option	Description
	<p>IDs for run-time components using the <a href="#">cc list inventory components</a> command.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p><b>Note:</b> Command Central only requires the <code>componentid</code> when you do not specify search criteria.</p> <p>When executing the command against a Platform Manager, specify the <code>componentid</code> before the action in the command syntax.</p> </div>
[ <i>criteria</i> ]	<p>Command Central only.</p> <p>Optional. Specifies to act only on the run-time components that match the search criteria you specify. For more information, see <a href="#">"Specifying Search Criteria for Lifecycle Commands"</a> on page 239.</p>
[ <i>options</i> ]	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands"</a> on page 59.</p>

### Usage Notes

- The command returns job information, that includes information such as the job ID and job status.
- You can execute the `cc get monitoring runtimestatus` command after executing the `cc exec lifecycle` command to determine when the requested action is complete. Use the `{--expected-values | -e}`, `{--check every | -c}`, and `{--wait | -w}` options with the `cc get monitoring runtimestatus` command to specify the results for which to check and how often to check for the results. For more information, see ["cc get monitoring"](#) on page 227.

### Examples When Executing on Command Central

- To start the run-time component on the installation with alias name "sag01" and component ID "OSGI-SPM":
 

```
cc exec lifecycle start sag01 OSGI-SPM
```

Because the `{--server | -s}`, `{--username | -u}`, and `{--password | -p}` options are not specified, the command uses the default server, user name, and password. For more information, see ["--server | -s"](#) on page 77, ["--username | -u"](#) on page 80, and ["--password | -p"](#) on page 76.
- To stop all run-time components that contain "OSGI" in the component display name:

```
cc exec lifecycle stop displayName=*OSGI* --password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see "[--server | -s](#)" on page 77 and "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.

### Examples When Executing on Platform Manager

- To restart the run-time component with ID "OSGI-SPM" that is installed in the installation managed by the Platform Manager with host name "rubicon2" and port "8092", using the authorization of the user with user name "Administrator" and password "manage":

```
cc exec lifecycle components OSGI-SPM restart --server
http://rubicon2:8092/spm --username Administrator --password manage
```

- To stop the run-time component with ID "OSGI-IS" that is installed in the installation managed by the Platform Manager with host name "rubicon2" and port "8092":

```
cc exec lifecycle components OSGI-IS stop
--server http://rubicon2:8092/spm --password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.

### Product-Specific Information

#### [Lifecycle Actions for Broker Server](#)

#### Related Commands

[cc list landscape nodes](#)

[cc list inventory components](#)

## Specifying Search Criteria for Lifecycle Commands

When executing the `cc exec lifecycle` command on a Command Central server, you can specify search criteria to identify the run-time components against the command should act. Supply the search criteria using the following format:

```
[logicalOperator=OR] attribute1=value1 attribute2=value2 ...
```

For the search criteria, you specify attribute values to match, for example `runtimeComponentId=OSGI-CCE`, where `runtimeComponentId` is the attribute and the value to match is `OSGI-CCE`.

### Specifying the Value

When specifying the value, you can include the `*` pattern-matching character to match multiple characters. For example, if you want to act only on run-time components with "cce" anywhere in their display names, use the following search criterion:

```
displayName=*cce*
```

**Important:** The search is case-sensitive.

### Attribute Names You Can Use in the Search Criteria

- nodeName
- nodeAlias
- nodeUrl
- environmentName
- environmentAlias
- runtimeComponentInfoId
- runtimeComponentId
- runtimeComponentDisplayName
- runtimeComponentProductId
- runtimeComponentCategory
- runtimeComponentRuntimeStatus
- runtimeComponentRuntimeParentId

You can combine any of the attribute names in this list with new search attributes added using the `cc create inventory components attributes` command. For more information about syntax and usage, see [cc create inventory components attributes](#).

### Logical Operators Used When Specifying Multiple Search Attributes

If you specify multiple search items, by default, the command performs an AND operation to return results that match all the specified criteria. For example, to act only on run-time components with “cce” anywhere in their display names *and* that are part of products that have the ID “OSGI”, use the following search criteria:

```
displayName=*cce* productId=OSGI
```

You can use an OR operation with two attributes. To do so, specify the `logicalOperator=OR` argument. For example, to act on run-time components with “cce” anywhere in their display names *or* that are part of products that have the ID “OSGI”, use the following search criteria:

```
displayName=*cce* logicalOperator=OR productId=OSGI
```

### Related Commands

[cc exec lifecycle](#)

[cc list inventory components](#)

---

# 10 Configuring Products

---

■ Configuring Instances .....	242
■ Testing a Configuration .....	243
■ References to File Locations in Product Configuration Files .....	243
■ Configuration Commands .....	244
■ Instance Management of Installed Products .....	277

## Configuring Instances

Perform the following procedure to add, edit, or delete items for a product instance configuration type over Command Central.

### To configure an instance

1. In the Environments pane, select the environment in which you want to configure a product instance.
2. Select the **Instances** tab.
3. Click the name of the instance you want to configure.
4. Select the **Configuration** tab.
5. From the list of available configuration types, select a configuration type.

Command Central displays the default or available values for the configuration data for the selected instance.

6. Configure the selected instance as follows:

To	Click
Add new data	
Edit data	To edit an item for a configuration type, click on the item that you want to update and click <b>Edit</b> .
Delete data	
Test whether data is added or edited successfully.	<b>Test</b>
For example, you can test new configuration data to perform a field-level validation before you save the configuration data.	

7. Click **Save** to save the configuration data.

---

## Testing a Configuration

---

While configuring an instance, you can test new configuration data before you save the configuration data for that instance.

When you test configuration data, a field-level input validation is done. If you have entered a port setting, the availability of the port is also tested.

---

### To test configuration data

1. Navigate to the **Environment > Instances > Configuration** tab of the product you want to configure.
2. Enter the configuration data.
3. Click **Test**.
4. Supply the appropriate data and click **Save**.

**Note:** If the test fails, Command Central displays a message that indicates the possible cause of the error. Resolve the error and try again until the test passes successfully.

---

## References to File Locations in Product Configuration Files

---

**Important:** Software AG recommends that you place all local files referenced in the configuration file of the product managed by Command Central in the *Software AG\_directory*.

When your product configuration refers to a local file on the managed product system, the type of path that you specify depends on the location of the referenced file.

- When the file is located in the *Software AG\_directory*, use relative paths. The relative path depends on how the managed product resolves relative locations and is typically relative to one of the following:
  - The product home installation directory
  - The product bin folder
  - *Software AG\_directory*
  - A supported location token
- When the file is not located in the *Software AG\_directory*, use absolute paths.

To minimize synchronization issues when using absolute paths, ensure that the absolute location is valid on all managed product systems. Note that when you

manage products in both Windows and UNIX environments, the absolute paths are usually different.

**Important:** When applying a snapshot template with configuration that relies on absolute paths, ensure that the target nodes run on the same operating system and have the same root installation directory.

### Migrating Product-Specific Configurations

Product configurations that are migrated from the source managed product system to the target managed product system do not automatically migrate the files referenced in the product configuration. Ensure that the referenced files are available on the target system at the referenced location.

## Configuration Commands

### cc get configuration common

Retrieves the schema for a specified configuration type.

#### Syntax

- Not supported by Command Central.
- Platform Manager Syntax:

```
cc get configuration common schema [options]
```

```
options :
[--debug | -d]
[--error | -r] file
[--log | -l] file
[--output | -o] file
[--password | -p] password
[--quiet | -q]
[--server | -s] url
[--username | -u] user_name
```

#### Arguments and Options

Argument or Option	Description
<i>schema</i>	<p>Required. Specifies the schema you want to retrieve. You can only retrieve schemas for common configuration types.</p> <p>The following list the schema names you can specify:</p> <ul style="list-style-type: none"> <li>■ CommonSettings.xsd</li> </ul>

Argument or Option	Description
	<ul style="list-style-type: none"> <li>■ EmailSettings.xsd</li> <li>■ JDBCSettings.xsd</li> <li>■ KeystoreTruststoreSettings.xsd</li> <li>■ LicenseLocation.xsd</li> <li>■ log4j.xsd</li> <li>■ PortSettings.xsd</li> </ul>
[options]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands"</a> on page 59.

### Usage Notes

- Configuration types that have IDs that start with “COMMON-”, for example COMMON-PORTS, are common configuration types that multiple products share. Common configuration types have normalized schemas that work for all products. However, these schemas still allow the following product-specific extensions:
  - You can have ExtendedProperties elements in the common schema XML files.
  - You can define common schema elements as optional.

Each product maps a common schema to its specific use. To learn how a product supports common configuration types and how a product’s configuration type is mapped to a common schema, use [cc get configuration data](#) to retrieve the data returned for a specific product’s configuration instance. The structure of the configuration data can vary based on the run-time component, product that owns the run-time product, and in some cases also based on the specific instance of a configuration type.

### Examples When Executing on Platform Manager

To execute a command on the Platform Manager server with host name “rubicon” and port “8090” to retrieve the “PortSettings.xsd” schema, using the authorization of the user with user name “Administrator” and password “manage”:

```
cc get configuration common PortSettings.xsd --server http://rubicon2:8092/spm -
-username Adminisrator --password manage
```

## cc get configuration compare

Compares the instances of a specified configuration type on two or more run-time components.

## Syntax

### ■ Command Central syntax:

```
cc get configuration compare configurationTypeId=typeid
runtimeComponentInfoId=id1 runtimeComponentInfoId=id2
[runtimeComponentInfoId=id3 ... runtimeComponentInfoId=idn] [options]
```

```
options:
[--accept | -a] content_type
[--debug | -d]
[--error | -r] file
[--format | -f] {tsv args | xml | csv args | json}
[--log | -l] file
[--output | -o] file
[--password | -p] password
[--quiet | -q]
[--server | -s] url]
[--username | -u] user_name
```

### ■ Not supported by Platform Manager

## Arguments and Options

Argument or Option	Description
<code>configurationTypeId=<i>typeid</i></code>	<p>Required. Specifies the configuration type to compare.</p> <p>You can determine the IDs for configuration types using <a href="#">cc list configuration types</a>.</p>
<code>runtimeComponentInfoId=<i>id1</i></code> <code>runtimeComponentInfoId=<i>id2</i></code> <code>[runtimeComponentInfoId=<i>id3</i></code> <code>...</code> <code>runtimeComponentInfoId=<i>idn</i>]</code>	<p>Required. Specifies the information IDs of two or more run-time components that you want to compare.</p> <p>The information ID is a combination of the alias name of the installation and the run-time component ID. For example, if the alias name of the installation is "sag1" and the run-time component ID is "OSGI-SPM", the information ID is "sag1-OSGI-SPM". You can view a list of installations and their alias names using <a href="#">cc list landscape nodes</a>. You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<code>[<i>options</i>]</code>	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options,</p>

Argument or Option	Description
	see <a href="#">"Options for the Commands"</a> on page 59.

## Usage Notes

- The command returns the results of the comparison.

## Example When Executing on Command Central

To execute a command on the Command Central server with host name "rubicon" and port "8090" to compare the instances of the configuration type with ID "COMMON-PORTS" that are configured on the run-time component that have the information IDs "sag1-OSGI-SPM1" and "sag1-OSGI-SPM2", using the authorization of the user with user name "Administrator" and password "manage", and have the information displayed on the console in XML format:

```
cc get configuration compare configurationTypeId=COMMON-PORTS
runtimeComponentInfoId=sag1-OSGI-SPM1 runtimeComponentInfoId=sag1-OSGI-SPM2
--format xml --server http://rubicon:8090/cce --username Administrator
--password manage
```

## cc create configuration data

Creates a new instance of a configuration type for a specified run-time component. For example, if you want to configure a new port for a Broker Server, use this command to supply the data for the configuration type COMMON-PORTS. By doing so, you create a new COMMON-PORTS instance.

## Syntax

- Command Central syntax:

```
cc create configuration data node_alias componentid typeid
[sharedsecret=text_string]
{--input | -i} file{.xml|.json|.properties} [options]
```

- Platform Manager syntax:

```
cc create configuration data componentid typeid
[sharedsecret=text_string]
{--input | -i} file{.xml|.json|.properties} [options]
```

```
options:
[{--debug | -d}]
[{--error | -r} file]
[{--log | -l} file]
[{--media-type | -m} content-type]
[{--password | -p} password]
[{--quiet | -q}]
[{--server | -s} url]
[{--username | -u} user_name]
```

## Arguments and Options

Argument or Option	Description
<code>node_alias</code>	<p>Command Central only.</p> <p>Required. Specifies the alias name of the installation in which the run-time component is installed.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<code>componentid</code>	<p>Required. Specifies the ID of the run-time component for which you want to create a new instance of a configuration type.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<code>typeid</code>	<p>Required. Specifies the ID of the configuration type that identifies the type of instance you want to create.</p> <p>You can determine the IDs for configuration types using <a href="#">cc list configuration types</a>.</p> <p>For information about the supported configuration types for a run-time component, see information in this reference for the product with which the run-time component is associated.</p>
<code>[sharedsecret=text_string]</code>	<p>Optional. Specify a shared secret password that Platform Manager uses to secure any passwords in the configuration data.</p> <p>If you omit this parameter, Platform Manager uses the shared secret defined for the environment or for the whole landscape. For more information, see "<a href="#">Working with Configurations that Include Passwords</a>" on page 156 and <a href="#">cc add security credentials sharedsecret</a>.</p>
<code>{--input   -i} file{.xml .json .properties}</code>	<p>Required. Identifies an input file that contains the configuration data. For more information, see "<a href="#">--input   -i</a>" on page 70.</p>

Argument or Option	Description
	<p><b>Note:</b>Based on the type of configuration instance you are attempting to create, all file types (.xml, .json, and .properties) might not be supported. Although not specifically supported, if you use plain text, the server attempts to convert the data into a supported format.</p> <p><b>Tip:</b> To determine how to specify the data in the input file, use <a href="#">cc get configuration data</a> to retrieve data for the same type of configuration instance you want to create. For example, if you want to use an XML file to specify the data to create an instance of a COMMON-PORTS configuration type, use <a href="#">cc get configuration data</a> with the <code>--format xml</code> option to retrieve the data for an existing COMMON-PORTS instance in XML format.</p>
[options]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands"</a> on page 59.

### Usage Notes

- Not all run-time components support the `cc create configuration data` command. For information about whether a run-time component supports a command, see information in this reference for the product with which the run-time component is associated.
- You can use [cc get configuration common](#) to validate input data that you want to use to create a configuration instance.
- You can retrieve schemas for common configuration types. You can use the schemas to validate an XML input data file. The schemas are available from a Platform Manager. For example, you might use the following to retrieve the log4j schema from a Platform Manager with host name "rubicon2" and port "8092":
 

```
http://rubicon:8092/spm/configuration/common/log4j.xsd
```

You can also use [cc get configuration common](#) to retrieve schemas for common configuration types.
- The output from the `cc create configuration data` command includes:
  - Instance ID of the new configuration instance

- Display name of the new instance
- Description of the new configuration instance
- ID of the associated configuration type
- ID of the run-time component
- URL of a physical configuration file if the data for the configuration instance is stored in a configuration file

### Example When Executing on Command Central

- The data to create a new instance of the COMMON-PORTS configuration type is in the c:\inputs\port\_data.xml file. To create the new configuration instance for the run-time component with the ID "OSGI-SPM" that is installed in the installation with alias name "sag01":

```
cc create configuration data sag01 OSGI-SPM COMMON-PORTS
--input c:\inputs\port_data.xml --password secret
```

- The data to create a new instance of the COMMON-PROXY configuration type is in the c:\inputs\proxy\_data.xml file. To create the new configuration instance for the run-time component with the ID "OSGI-SPM" that is installed in the installation with alias name "sag01" and use "mysecret123" as the shared secret to secure the proxy configuration password:

```
cc create configuration data sag01 OSGI-SPM COMMON-PROXY
sharedsecret=mysecret123 --input
c:\inputs\proxy_data.xml --password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see "[--server | -s](#)" on page 77 and "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.

### Example When Executing on Platform Manager

The data to create a new instance of the COMMON-PORTS configuration type is in the c:\inputs\port\_data.xml file. To create the new configuration instance for the run-time component that has the ID "OSGI-SPM" and is managed by the Platform Manager with host name "rubicon2" and port "8092":

```
cc create configuration data OSGI-SPM COMMON-PORTS --input
c:\inputs\port_data.xml --server http://rubicon2:8092/spm
--password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.

### Product-Specific Information

[Configuration Types that webMethods Broker Supports](#)

[Configuration Types that IntegrationServer-instanceName Supports](#)

[Configuration Types that My webMethods Server-ENGINE Supports](#)

## Related Commands

[cc delete configuration data](#)

[cc delete configuration data](#)

[cc get configuration data](#)

[cc get configuration instances](#)

[cc list configuration instances](#)

[cc get configuration types](#)

[cc list configuration types](#)

[cc list inventory components](#)

## cc delete configuration data

Deletes a configuration instance from a specified run-time component. For example, you might want to delete a previously created port on Integration Server.

### Syntax

#### ■ Command Central syntax:

```
cc delete configuration data node_alias componentid instanceid [options]
```

#### ■ Platform Manager syntax:

```
cc delete configuration data componentid instanceid [options]
```

```
options :
[--debug | -d]
[--error | -r] file
[--force]
[--log | -l] file
[--password | -p] password
[--quiet | -q]
[--server | -s] url
[--username | -u] user_name
```

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	<p>Command Central only.</p> <p>Required. Specifies the alias name of the installation in which the run-time component is installed.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>

Argument or Option	Description
<i>componentid</i>	<p>Required. Specifies the ID of the run-time component from which you want to delete a configuration instance.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<i>instanceid</i>	<p>Required. Specifies the ID of the instance you want to delete.</p> <p>You can determine the IDs for configuration instances using <a href="#">cc list configuration instances</a>.</p>
[ <i>options</i> ]	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands"</a> on page 59.</p>

### Usage Notes

- You cannot delete some configuration data. In general, if you can create the configuration data, you can also usually delete it. For example, you can add a new Integration Server port, and you can also delete an Integration Server port. However, you cannot create new ports on Broker Server, and you cannot delete Broker Server ports. The command returns an error if you attempt to perform an unsupported operation.
- You can use [cc exec configuration validation delete](#) to determine whether it is valid to delete the configuration instance.

### Example When Executing on Command Central

To delete the configuration instance with ID "COMMON-PORTS-com.softwareag.sshd.pid.properties" from the run-time component with ID "OSGI-SPM", which is installed in the installation with alias name "sag01" using the authorization of the user with user name "Administrator" and password "manage":

```
cc delete configuration data sag01 OSGI-SPM COMMON-PORTS-
com.softwareag.sshd.pid.properties --username Administrator
--password manage
```

Because the {`--server | -s`} option is not specified, the command uses the default server. For more information, see ["--server | -s"](#) on page 77.

### Example When Executing on Platform Manager

To delete the configuration instance with ID "COMMON-PORTS-com.softwareag.sshd.pid.properties" from the run-time component that has the ID "OSGI-SPM" and is managed by the Platform Manager with host name "rubicon2" and

port “8092”, and execute the command with the authorization of the user with user name “Administrator” and password “manage”:

```
cc delete configuration data OSGI-SPM COMMON-PORTS-
com.softwareag.sshd.pid.properties --server http://rubicon2:8092/spm
--username Administrator --password manage
```

### Related Commands

[cc create configuration data](#)

[cc create configuration data](#)

[cc get configuration data](#)

[cc update configuration data](#)

[cc list configuration instances](#)

[cc list inventory components](#)

## cc get configuration data

Retrieves data for a specified configuration instance that belongs to a specified runtime component. For example, you might want to retrieve the data for an instance of a configured port. The retrieved data for an instance of a port might include whether the port is enabled, the port number, and the port’s protocol.

### Syntax

#### ■ Command Central syntax:

```
cc get configuration data node_alias componentid instanceid
[sharedsecret=text_string] [options]
```

#### ■ Platform Manager syntax:

```
cc get configuration data componentid instanceid
[sharedsecret=text_string] [options]
```

```
options :
[{--accept | -a} content_type]
[{--debug | -d}]
[{--error | -r} file]
[{--format | -f} {text | xml | json}]
[{--log | -l} file]
[{--output | -o} file]
[{--password | -p} password]
[{--quiet | -q}]
[{--server | -s} url]
[{--username | -u} user_name]
```

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	Command Central only.

Argument or Option	Description
	<p>Required. Specifies the alias name of the installation in which the run-time component is installed.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<i>componentid</i>	<p>Required. Specifies the ID of the run-time component for which you want to retrieve instance data.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<i>instanceid</i>	<p>Required. Specifies the ID of the instance for which you want to retrieve data.</p> <p>You can determine the IDs for configuration instances using <a href="#">cc list configuration instances</a>.</p>
<i>[sharedsecret=text_string]</i>	<p>Optional. Specify a shared secret password that Platform Manager uses to secure any passwords in the configuration data that you want to retrieve.</p> <p>If you omit this parameter, Platform Manager retrieves the configuration data using the shared secret defined for the environment or for the whole landscape. For more information, see <a href="#">"Working with Configurations that Include Passwords" on page 156</a> and <a href="#">cc add security credentials sharedsecret</a>. If Platform Manager cannot find a shared secret that matches the shared secret specified when creating the configuration instance, Platform Manager does not include the passwords that the configuration contains in the output.</p> <p>When you omit this parameter:</p> <ul style="list-style-type: none"> <li>■ If the command is executed on Command Central, Platform Manager retrieves the configuration data using the shared secret defined for the environment or for the whole landscape. For more information, see <a href="#">"Working with Configurations that Include Passwords" on page 156</a> and <a href="#">cc add security credentials sharedsecret</a>.</li> <li>■ If the command is executed on Platform Manager, Platform Manager does not include the</li> </ul>

Argument or Option	Description
	passwords that the configuration contains in the output.
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

- Use [cc get configuration instances](#) or [cc list configuration instances](#) if you want information about an instance, such as the instance ID, the display name for an instance, and the description for an instance, rather than the data for an instance.
- If you do not specify the `{--format | -f}` option, the default output format is based on from where you execute the command:
  - If you execute the command from the command line, a batch script, or a shell script, the default format is plain text format.
  - If you execute the command from an Ant script, the default format is XML format.

### Example When Executing on Command Central

- To execute a command on the Command Central server with host name "rubicon" and port "8090" to retrieve instance data for the configuration instance with ID "COMMON-PORTS-com.softwareag.sshd.pid.properties" that belongs to the run-time component that has the ID "OSGI-SPM" and runs in the installation with alias name "sag01", using the authorization of the user with user name "Administrator" and password "manage", and have the information displayed on the console in XML format:

```
cc get configuration data sag01 OSGI-SPM COMMON-PORTS-
com.softwareag.sshd.pid.properties --format xml
--server http://rubicon:8090/cce --username Administrator
--password manage
```

- To retrieve instance data for the configuration instance with ID "COMMON-PROXY" that belongs to the run-time component with ID "OSGI-SPM" and runs in the installation with alias name "sag01", using "mysecret123" as the shared secret to secure the proxy configuration password, and save the configuration data with the encrypted password in the "configWithEncryptedData.xml" file:

```
cc get configuration data sag01 OSGI-SPM COMMON-PROXY
sharedsecret=mysecret123 --output
configWithEncryptedData.xml
```

### Example When Executing on Platform Manager

To retrieve instance data for the configuration instance with ID "COMMON-PORTS-com.softwareag.sshd.pid.properties" that belongs to the run-time component that

has the ID “OSGI-SPM” and is managed by the Platform Manager with host name “rubicon2” and port “8092”, and have the information returned to the output file “port\_data” in XML format:

```
cc get configuration data OSGI-SPM COMMON-PORTS-
com.softwareag.sshd.pid.properties --output port_data --format xml
--server http://rubicon2:8092/spm
```

Because the `{--username | -u}` and `{--password | -p}` options are not specified, the command uses the default user name and password. For more information, see “[--username | -u](#)” on page 80 and “[--password | -p](#)” on page 76.

## Related Commands

[cc create configuration data](#)

[cc delete configuration data](#)

[cc update configuration data](#)

[cc get configuration instances](#)

[cc list configuration instances](#)

[cc get configuration types](#)

[cc list configuration types](#)

[cc list inventory components](#)

## cc update configuration data

Updates the data for a specified configuration instance that belongs to a specified run-time component. For example, you might want to update the port number of a COMMON-PORTS configuration instance.

### Syntax

#### ■ Command Central syntax:

```
cc update configuration data node_alias componentid instanceid
[sharedsecret=text_string]
[--input | -i] filename{.xml|.json|.properties} [options]
```

#### ■ Platform Manager syntax:

```
cc update configuration data componentid instanceid
[sharedsecret=text_string]
[--input | -i] filename{.xml|.json|.properties} [options]
```

```
options: [--debug | -d]
[--error | -r] file
[--log | -l] file
[--media-type | -m] content-type
[--password | -p] password
[--quiet | -q]
[--server | -s] url
[--username | -u] user_name
```

## Arguments and Options

Argument or Option	Description
<i>node_alias</i>	<p>Command Central only.</p> <p>Required. Specifies the alias name of the installation in which the run-time component is installed.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<i>componentid</i>	<p>Required. Specifies the ID of the run-time component to which the instance you want to update belongs.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<i>instanceid</i>	<p>Required. Specifies the ID of the instance for which you want to update data.</p> <p>You can determine the IDs for configuration instances using <a href="#">cc list configuration instances</a>.</p>
[sharedsecret= <i>text_string</i> ]	<p>Optional. Specify a shared secret password that Platform Manager uses to secure any passwords in the configuration data that you want to update.</p> <p>If you omit this parameter, Platform Manager uses the shared secret defined for the environment or for the whole landscape. For more information, see <a href="#">"Working with Configurations that Include Passwords"</a> on <a href="#">page 156</a> and <a href="#">cc add security credentials sharedsecret</a>.</p>
{--input   -i} <i>filename</i> {.xml  .json .properties}	<p>Required. Identifies an input file that contains the updated configuration data. For more information, see <a href="#">"--input   -i"</a> on <a href="#">page 70</a>.</p> <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p><b>Note:</b>Based on the type of configuration instance you are attempting to create, all file types (.xml, .json, and .properties) might not be supported.</p> </div>

Argument or Option	Description
	<p>When updating instances of common configuration types, XML, json, and properties are all supported types of the input file. However, all of these file types might not be supported when creating instances of a product-specific configuration type.</p> <div data-bbox="721 548 1373 957" style="background-color: #f0f0f0; padding: 10px;"> <p><b>Tip:</b> To determine how to specify the data in the input file, use <code>cc get configuration data</code> to retrieve data for the configuration instance you want to update. For example, if you want to use an XML file to specify the data to update an instance of a COMMON-PORTS configuration type, use <code>cc get configuration data</code> with the <code>--format xml</code> option to retrieve the data for the COMMON-PORTS instance in XML format.</p> </div>
[options]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Usage Notes

- The data in the input file must match the expected schema for the configuration type.
- You can use `cc exec configuration validation update` to validate input data that you want to use to update the configuration instance.
- You can retrieve schemas for common configuration types. You can use the schemas to validate an XML input data file. The schemas are available from a Platform Manager. For example, you might use the following to retrieve the log4j schema from a Platform Manager with host name "rubicon2" and port "8092":

```
http://rubicon:8092/spm/configuration/common/log4j.xsd
```

You can also use `cc get configuration common` to retrieve schemas for common configuration types.

- You can retrieve schemas for common configuration types. The schemas are available from a Platform Manager using the following where *hostname* is the host name of a Platform Manager server and *port* is its port number:

```
http://hostname:port/spm/configuration/common/
```

You can also use `cc get configuration common` to retrieve schemas for common configuration types.

- You can specify the content type for the output data for the command either in the Accept header of the REST client or by adding the `--output-format` option. For example, if you add `--output-format text`, the command displays data on the console in plain text format. For more information, see "`--output-format | -f`" on page 76.

If you do not specify the content type for the output data, the command returns the default output format, XML.

- When using an XML input data file to update a configuration instance, use the `cc get configuration data` command with the `--output | -o filename.xml` option to retrieve a copy of the XML input data file for the configuration instance you require.

After including the required updated data in the copy of the XML data file, you can use the `cc update configuration data` command to update the configuration instance.

### Examples When Executing on Command Central

- The data to update a COMMON-PORTS instance is in the `c:\inputs\port_data.xml` file. To update the configuration instance with ID "COMMON-PORTS-com.softwareag.sshd.pid.properties" for the run-time component with ID "OSGI-SPM", which is installed in the installation with alias name "sag01" using the authorization of the user with user name "Administrator" and password "manage":

```
cc update configuration data sag01 OSGI-SPM COMMON-PORTS-
com.softwareag.sshd.pid.properties --input c:\inputs\port_data.xml
--username Administrator --password manage
```

- The `Software AG_directory/profiles/CCE/configuration/security/user.txt` file is updated with a new password that replaces the default Command Central administrator password. To update the modified `user.txt` file for the instance with ID "ENGINE SIN-INTERNAL-USERS-users.txt" and the run-time component with ID "OSGI-SPM", which is installed in the installation with alias name "sag01":

```
cc update configuration data sag01 OSGI-SPM-ENGINE SIN-INTERNAL-USERS-users.txt
--input
Software AG_directory
/profiles/CCE/
configuration/security/users.txt
```

- To update instance data for the configuration instance with ID "COMMON-PROXY" that belongs to the run-time component with ID "OSGI-SPM" and runs in the installation with alias name "sag01", using "mysecret123" as the shared secret to secure the proxy configuration password, and save the updated configuration data with the encrypted password in the "configWithEncryptedData.xml" file:

```
cc update configuration data sag01 OSGI-SPM COMMON-PROXY
sharedsecret=mysecret123 --input
configWithEncryptedData.xml
```

Because the `{--server | -s}` option is not specified, the command uses the default server. For more information, see "[--server | -s](#)" on page 77.

### Example When Executing on Platform Manager

The data to update a COMMON-PORTS instance is in the `c:\inputs\port_data.xml` file. To update the configuration instance with ID "COMMON-PORTS-com.softwareag.sshd.pid.properties" for the run-time component that has the ID "OSGI-SPM" and is managed by the Platform Manager with host name "rubicon2" and port "8092":

```
cc update configuration data OSGI-SPM COMMON-PORTS-  
com.softwareag.sshd.pid.properties --input c:\inputs\port_data.xml  
--server http://rubicon2:8092/spm --password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.

### Product-Specific Information

[Configuration Types that webMethods Broker Supports](#)

[Configuration Types that IntegrationServer-instanceName Supports](#)

[Configuration Types that My webMethods Server-ENGINE Supports](#)

### Related Commands

[cc create configuration data](#)

[cc create configuration data](#)

[cc delete configuration data](#)

[cc get configuration data](#)

[cc get configuration instances](#)

[cc list configuration instances](#)

[cc get configuration types](#)

[cc list configuration types](#)

[cc list inventory components](#)

## cc get configuration instances

Retrieves information about a specific configuration instance that belongs to a specified run-time component. For example, you might want to retrieve information about an instance of a configuration properties file. Information about a configuration instance can include:

- Instance ID

- Type ID for the configuration type associated with the instance
- Display name for the instance
- Description of the instance
- URL providing the location of the configuration instance
- The ID of the run-time component to which the instance belongs

## Syntax

- Command Central syntax:

```
cc get configuration instances node_alias componentid instanceid
[options]
```

- Platform Manager syntax:

```
cc get configuration instances componentid instanceid [options]
```

```
options:
[--accept | -a] content_type
[--debug | -d]
[--error | -r] file
[--format | -f] {tsv args | text | xml | csv args | json}
[--log | -l] file
[--output | -o] file
[--quiet | -q]
[--password | -p] password
[--server | -s] url
[--username | -u] user_name
```

## Arguments and Options

Argument or Option	Description
<i>node_alias</i>	<p>Command Central only.</p> <p>Required. Specifies the alias name of the installation in which the run-time component is installed.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<i>componentid</i>	<p>Required. Specifies the ID of the run-time component for which you want to retrieve instance information.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<i>instanceid</i>	<p>Required. Specifies the ID of the instance for which you want to retrieve information.</p> <p>You can determine the IDs for configuration instances using <a href="#">cc list configuration instances</a>.</p>

Argument or Option	Description
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Usage Notes

- To retrieve the data for a specific instance, use [cc get configuration data](#).

### Example When Executing on Command Central

To execute a command on the Command Central server with host name "rubicon" and port "8090" to retrieve information about the configuration instance with ID "COMMON-PORTS-com.softwareag.sshd.pid.properties" that belongs to the run-time component that has the ID "OSGI-SPM" and runs in the installation with alias name "sag01", using the authorization of the user with user name "Administrator" and password "manage", and have the information displayed on the console in XML format:

```
cc get configuration instances sag01 OSGI-SPM COMMON-PORTS-
com.softwareag.sshd.pid.properties --format xml
--server http://rubicon:8090/cce --username Administrator
--password manage
```

### Example When Executing on Platform Manager

To retrieve information about the configuration instance with ID "COMMON-PORTS-com.softwareag.sshd.pid.properties" that belongs to the run-time component that has the ID "OSGI-SPM" and is managed by the Platform Manager with host name "rubicon2" and port "8092", and have the information displayed on the console in XML format:

```
cc get configuration instances OSGI-SPM COMMON-PORTS-
com.softwareag.sshd.pid.properties --format xml --server
http://rubicon2:8092/spm
```

Because the `{--username | -u}` and `{--password | -p}` options are not specified, the command uses the default user name and password. For more information, see "[--username | -u](#)" on page 80 and "[--password | -p](#)" on page 76.

### Related Commands

[cc get configuration data](#)

[cc list configuration instances](#)

[cc get configuration types](#)

[cc list configuration types](#)

[cc list inventory components](#)

## cc list configuration instances

Lists the configuration instances that belongs to a specified run-time component. Information about a configuration instance can include:

- Instance ID
- Type ID for the configuration type associated with the instance
- Display name for the instance
- Description of the instance
- URL providing the location of the configuration instance
- The ID of the run-time component to which the instance belongs

### Syntax

- Command Central syntax:

```
cc list configuration instances node_alias componentid [instanceid]
[options]
```

- Platform Manager syntax:

```
cc list configuration instances componentid [instanceid] [options]
```

```
options:
[--accept | -a] content_type]
[--debug | -d]
[--error | -r] file]
[--format | -f] {tsv args | text | xml | csv args | json}]
[--log | -l] file]
[--output | -o] file]
[--password | -p] password]
[--quiet | -q]
[--server | -s] url]
[--username | -u] user_name]
```

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	<p>Command Central only.</p> <p>Required. Specifies the alias name of the installation in which the run-time component is installed.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<i>componentid</i>	<p>Required. Specifies the ID of the run-time component for which you want to list configuration instances.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>

Argument or Option	Description
[ <i>instanceid</i> ]	Optional. Specifies the ID of the instance for which you want to retrieve information. If you do not specify an instance ID, the command lists information for all instances that belong to the run-time component identified by the <i>componentid</i> argument.
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

- To retrieve the data for a specific instance, use [cc get configuration data](#).

### Example When Executing on Command Central

To execute a command on the Command Central server with host name "rubicon" and port "8090" to list configuration instances that belong to the run-time component that has the ID "OSGI-SPM" and runs in the installation with alias name "sag01", using the authorization of the user with user name "Administrator" and password "manage", and have the information returned to the output file "config\_instances" in XML format:

```
cc list configuration instances sag01 OSGI-SPM --format xml --output
config_instances --server http://rubicon:8090/cce
--username Administrator --password manage
```

### Example When Executing on Platform Manager

To list configuration instances that belong to the run-time component that has the ID "OSGI-SPM" and is managed by the Platform Manager with host name "rubicon2" and port "8092", using the authorization of the user with user name "Administrator" and password "manage", and have the information returned to the console in JavaScript Object Notation format:

```
cc list configuration instances OSGI-SPM --format json --server
http://rubicon2:8092/spm --username Administrator --password manage
```

### Related Commands

[cc get configuration data](#)

[cc get configuration instances](#)

[cc get configuration types](#)

[cc list configuration types](#)

[cc list inventory components](#)

## cc get configuration types

Retrieves information for a specified configuration type associated with a specified run-time component. Information about a configuration type can include:

- Type ID
- Display name if one is assigned; otherwise null
- Description if one is assigned; otherwise null
- Content type of the data

### Syntax

- Command Central syntax:

```
cc get configuration types node_alias componentid typeid [options]
```

- Platform Manager syntax:

```
cc get configuration types componentid typeid [options]
```

```
options:
[--accept | -a] content_type ]
[--debug | -d]
[--error | -r] file ]
[--format | -f] {tsv args | text | xml | csv args | json} ]
[--log | -l] file ]
[--output | -o] file ]
[--password | -p] password ]
[--quiet | -q]
[--server | -s] url ]
[--username | -u] user_name ]
```

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	<p>Command Central only.</p> <p>Required. Specifies the alias name of the installation in which the run-time component is installed.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<i>componentid</i>	<p>Required. Specifies the ID of the run-time component for which you want to retrieve a configuration type.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<i>typeid</i>	<p>Required. Specifies the ID of the configuration type for which you want to retrieve information.</p>

Argument or Option	Description
	You can determine the IDs for configuration types using <a href="#">cc list configuration types</a> .
<code>[options]</code>	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

- Run-time components support a set of configuration types. Use [cc list configuration types](#) to learn what configuration types that a run-time component supports.

### Example When Executing on Command Central

To execute a command on the Command Central server with host name "rubicon" and port "8090" to retrieve information about the configuration type with ID "COMMON-PORTS" that is associated with run-time component that has the ID "OSGI-SPM" and runs in the installation with alias name "sag01", using the authorization of the user with user name "Administrator" and password "manage", and have the information displayed on the console in XML format:

```
cc get configuration types sag01 OSGI-SPM COMMON-PORTS --format xml
--server http://rubicon:8090/cce --username Administrator
--password manage
```

### Example When Executing on Platform Manager

To retrieve information about the configuration type with ID "COMMON-PORTS" that is associated with run-time component that has the ID "OSGI-SPM" and is managed by the Platform Manager with host name "rubicon2" and port "8092", and have the output displayed on the console using the default format:

```
cc get configuration types OSGI-SPM COMMON-PORTS --server
http://rubicon2:8092/spm --password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see ["--username | -u" on page 80](#). The command specifies "secret" for the user's password.

### Product-Specific Information

[Configuration Types that webMethods Broker Supports](#)

[Configuration Types that IntegrationServer-instanceName Supports](#)

[Configuration Types that My webMethods Server-ENGINE Supports](#)

## Related Commands

[cc create configuration data](#)

[cc get configuration data](#)

[cc update configuration data](#)

[cc get configuration instances](#)

[cc list configuration instances](#)

[cc list configuration types](#)

[cc list inventory components](#)

## cc list configuration types

Lists information about configuration types for the specified run-time component. A run-time component can support both common configuration types, such as ports, logs, and licenses, and/or custom configuration types that are specific to the run-time component. Information about a configuration type can include:

- Type ID
- Display name if one is assigned; otherwise null
- Description if one is assigned; otherwise null
- Content type of the data

### Syntax

- Command Central syntax:

```
cc list configuration types node_alias componentid [typeid] [options]
```

- Platform Manager syntax:

```
cc list configuration types componentid [typeid] [options]
```

```

options:
[--accept | -a] content_type]
[--debug | -d]
[--error | -r] file]
[--format | -f] {tsv args | text | xml | csv args | json}]
[--log | -l] file]
[--output | -o] file]
[--password | -p] password]
[--quiet | -q]
[--server | -s] url]
[--username | -u] user_name]

```

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	Command Central only.

Argument or Option	Description
	<p>Required. Specifies the alias name of the installation in which the run-time component is installed.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<i>componentid</i>	<p>Required. Specifies the ID of the run-time component for which you want to list configuration types.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
[ <i>typeid</i> ]	<p>Optional. Specifies the ID of the configuration type for which you want to retrieve information. If you do not specify a type ID, the command lists information for all configuration types for the run-time component identified by the <i>componentid</i> argument.</p>
[ <i>options</i> ]	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a>.</p>

### Usage Notes

- Configuration types that have IDs that start with “COMMON-”, for example COMMON-PORTS, are common configuration types that multiple products share. Common configuration types have normalized schemas that work for all products. However, these schemas still allow product-specific extensions:
  - Having ExtendedProperties elements in the common schema XML files
  - Defining common schema elements as optional.

Each product maps a common schema to its specific use. To learn how a product supports common configuration types and how a product’s configuration type is mapped to a common schemas, use [cc get configuration data](#) to retrieve the data returned for a specific product’s configuration instance. The structure of the configuration data can vary based on the run-time component, product that owns the run-time product, and in some cases also based on the specific instance of a configuration type.

### Example When Executing on Command Central

To execute a command on the Command Central server with host name “rubicon” and port “8090” to list the configuration types for the run-time component that has the

ID “OSGI-SPM” and is running in the installation with alias name “sag01”, using the authorization of the user with user name “Administrator” and password “manage”, and have the information returned to the output file “config\_types” in XML format:

```
cc list configuration types sag01 OSGI-SPM --format xml
--output config_types --server http://rubicon:8090/cce
--username Administrator --password manage
```

### Example When Executing on Platform Manager

To list the configuration types for the run-time component that has the ID “OSGI-SPM” and is managed by the Platform Manager with host name “rubicon2” and port “8092”, using the authorization of the user with user name “Administrator” and password “manage”, and have the information returned to the console in JavaScript Object Notation format:

```
cc list configuration types OSGI-SPM --format json --server
http://rubicon2:8092/spm --username Administrator --password manage
```

### Product-Specific Information

[Configuration Types that webMethods Broker Supports](#)

[Configuration Types that IntegrationServer-instanceName Supports](#)

[Configuration Types that My webMethods Server-ENGINE Supports](#)

### Related Commands

[cc create configuration data](#)

[cc get configuration data](#)

[cc update configuration data](#)

[cc get configuration instances](#)

[cc list configuration instances](#)

[cc get configuration types](#)

[cc list inventory components](#)

## cc exec configuration validation create

Validates the configuration instance data in the supplied input file. If the input data is valid, you can then use [cc create configuration data](#) to create a configuration instance.

### Syntax

#### ■ Command Central syntax:

```
cc exec configuration validation node_alias componentid create typeid
--input | -i} filename{.xml|.json|.properties} [options]
```

#### ■ Platform Manager syntax:

```

cc exec configuration validation componentid create typeid
  {--input | -i} filename{.xml|.json|.properties} [options]

options:
[--debug | -d]
[--error | -r} file]
[--log | -l} file]
[--password | -p} password]
[--quiet | -q]
[--server | -s} url]
[--username | -u} user_name]

```

## Arguments and Options

Argument or Option	Description
<i>node_alias</i>	<p>Command Central only.</p> <p>Required. Specifies the alias name of the installation in which the run-time component is installed.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<i>componentid</i>	<p>Required. Specifies the ID of the run-time component for which you want to validate instance data that you might want to use to create a new configuration type.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<i>typeid</i>	<p>Required. Specifies the ID of the configuration type that identifies the type of instance data you want to validate.</p> <p>You can determine the IDs for configuration types using <a href="#">cc list configuration types</a>.</p> <p>For information about the supported configuration types for a run-time component, see information in this reference for the product with which the run-time component is associated.</p>
<i>--input</i>   -i} <i>filename</i> {.xml  .json .properties}	<p>Required. Identifies an input file that contains the configuration data to validate. For more information, see "<a href="#">--input   -i</a>" on <a href="#">page 70</a>.</p> <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p><b>Note:</b> Based on the type of configuration data you are attempting to validate, all file types (.xml, .json, and .properties) might not</p> </div>

Argument or Option	Description
	<p>be supported. Although not specifically supported, if you use plain text, the server attempts to convert the data into a supported format.</p> <p><b>Tip:</b> To determine how to specify the data in the input file, use <code>cc get configuration data</code> to retrieve data for the same type of configuration instance you want to validate. For example, if you want to use an XML file for configuration data for a COMMON-PORTS configuration type, use <code>cc get configuration data</code> with the <code>--format xml</code> option to retrieve the data for an existing COMMON-PORTS instance in XML format.</p>
<code>[options]</code>	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a>.</p>

### Usage Notes

- Not all run-time components support the `cc exec configuration validation create` command. For information about whether a run-time component supports a command, see information in this reference for the product with which the run-time component is associated.
- Use this command to determine whether data for a new configuration instance is valid. This command does not create a new configuration instance. If the data in the input file is valid, you create a new configuration instance using the data by executing `cc create configuration data` command and supplying the validated input file.
- The `cc exec configuration validation create` command outputs either no messages or informational, warning, and/or error messages.
  - When the command outputs no messages or only informational and warning messages, the input data is valid. You can use the data with the `cc create configuration data` command to create a configuration instance.
  - When the command outputs error messages, the input data is not valid. The `cc create configuration data` command will fail if you use the data to attempt to create a configuration instance.

### Example When Executing on Command Central

The data for a COMMON-PORTS configuration type instance is in the `c:\inputs\port_data.xml` file. To validate the instance data for the run-time component with the ID "OSGI-SPM" that is installed in the installation with alias name "sag01":

```
cc exec configuration validation sag01 OSGI-SPM create COMMON-PORTS
--input c:\inputs\port_data.xml --password secret
```

Because the `{--server | -s}` and `{--username | -u}` options are not specified, the command uses the default server and user name. For more information, see "[--server | -s](#)" on page 77 and "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.

### Example When Executing on Platform Manager

The data for a COMMON-PORTS configuration type instance is in the `c:\inputs\port_data.xml` file. To validate the instance data for the run-time component with the ID "OSGI-SPM" and is managed by the Platform Manager with host name "rubicon2" and port "8092":

```
cc exec configuration validation OSGI-SPM create COMMON-PORTS
--input c:\inputs\port_data.xml --server http://rubicon2:8092/spm
--password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.

### Product-Specific Information

[Configuration Types that IntegrationServer-instanceName Supports](#)

[Configuration Types that My webMethods Server-ENGINE Supports](#)

### Related Commands

[cc create configuration data](#)

[cc exec configuration validation delete](#)

[cc exec configuration validation delete](#)

## cc exec configuration validation delete

Determines whether a configuration instance can be deleted. If check is successful, you can then use [cc delete configuration data](#) to delete the configuration instance.

### Syntax

- Command Central syntax:

```
cc exec configuration validation node_alias componentid delete instanceid
[options]
```

- Platform Manager syntax:

```
cc exec configuration validation componentid delete instanceid [options]
```

```
options:
[{--debug | -d}]
[{--error | -r} file]
[{--log | -l} file]
```

```
[{--password | -p} password]
[{-quiet | -q}]
[{-server | -s} url]
[{-username | -u} user_name]
```

## Arguments and Options

Argument or Option	Description
<i>node_alias</i>	<p>Command Central only.</p> <p>Required. Specifies the alias name of the installation in which the run-time component is installed.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<i>componentid</i>	<p>Required. Specifies the ID of the run-time component that owns the instance.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<i>instanceid</i>	<p>Required. Specifies the ID of the instance you want to check.</p> <p>You can determine the IDs for configuration instances using <a href="#">cc list configuration instances</a>.</p>
[ <i>options</i> ]	<p>Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands"</a> on page 59.</p>

## Usage Notes

- Use this command to determine whether you can delete a configuration instance. This command does not delete the configuration instance.
- The `cc exec configuration validation delete` command outputs either no messages or informational, warning, and/or error messages.
  - When the command outputs no messages or only informational and warning messages, the check is successful. You can use the [cc delete configuration data](#) command to delete the configuration instance.
  - When the command outputs error messages, the check failed. The [cc delete configuration data](#) command will fail if attempt to delete the configuration instance.

### Example When Executing on Command Central

To check whether you can delete the configuration instance with ID “COMMON-PORTS-com.softwareag.sshd.pid.properties” from the run-time component with ID “OSGI-SPM”, which is installed in the installation with alias name “sag01” using the authorization of the user with user name “Administrator” and password “manage”:

```
cc delete configuration data sag01 OSGI-SPM COMMON-PORTS-
com.softwareag.sshd.pid.properties --username Administrator
--password manage
```

Because the `{--server | -s}` option is not specified, the command uses the default server. For more information, see “[--server | -s](#)” on page 77.

### Example When Executing on Platform Manager

To check whether you can delete the configuration instance with ID “COMMON-PORTS-com.softwareag.sshd.pid.properties” from the run-time component that has the ID “OSGI-SPM” and is managed by the Platform Manager with host name “rubicon2” and port “8092”, and execute the command with the authorization of the user with user name “Administrator” and password “manage”:

```
cc delete configuration data OSGI-SPM COMMON-PORTS-
com.softwareag.sshd.pid.properties -server http://rubicon2:8092/spm
--username Administrator --password manage
```

### Related Commands

[cc create configuration data](#)

[cc exec configuration validation delete](#)

[cc exec configuration validation delete](#)

## cc exec configuration validation update

Validates the configuration instance data in the supplied input file to determine whether you can use it to update a specified configuration instance. If the input data is valid, you can then use [cc delete configuration data](#) to update the configuration instance.

### Syntax

#### ■ Command Central syntax:

```
cc exec configuration validation node_alias componentid update instanceid
{--input | -i} filename{.xml|.json|.properties} [options]
```

#### ■ Platform Manager syntax:

```
cc exec configuration validation componentid update instanceid
{--input | -i} filename{.xml|.json|.properties} [options]
```

```
options :
{--debug | -d}
{--error | -r} file
{--log | -l} file
{--password | -p} password
```

```
[{--quiet | -q}]
[{--server | -s} url]
[{--username | -u} user_name]
```

## Arguments and Options

Argument or Option	Description
<i>node_alias</i>	<p>Command Central only.</p> <p>Required. Specifies the alias name of the installation in which the run-time component is installed.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<i>componentid</i>	<p>Required. Specifies the ID of the run-time component for which you want to validate instance data that you might want to use to update a configuration instance.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<i>instanceid</i>	<p>Required. Specifies the ID of the instance.</p> <p>You can determine the IDs for configuration instances using <a href="#">cc list configuration instances</a>.</p>
<pre>{--input   -i} filename{.xml  .json .properties}</pre>	<p>Required. Identifies an input file that contains the configuration data to validate. For more information, see "<a href="#">--input   -i</a>" on <a href="#">page 70</a>.</p> <div style="background-color: #f0f0f0; padding: 10px;"> <p><b>Note:</b> Based on the type of configuration data you are attempting to validate, all file types (.xml, .json, and .properties) might not be supported. Although not specifically supported, if you use plain text, the server attempts to convert the data into a supported format.</p> <p><b>Tip:</b> To determine how to specify the data in the input file, use <a href="#">cc get configuration data</a> to retrieve data for the same type of configuration instance you want to validate. For example, if you want to use an XML file for configuration data for a COMMON-PORTS configuration type, use <a href="#">cc get configuration data</a> with the <code>--format xml</code> option to retrieve the data for an</p> </div>

Argument or Option	Description
	existing COMMON-PORTS instance in XML format.
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

- Use this command to determine whether data to update a configuration instance is valid. This command does not update the configuration instance. If the data in the input file is valid, you update the configuration instance using the data by executing `cc delete configuration data` command and supplying the validated input file.
- The `cc exec configuration validation update` command outputs either no messages or informational, warning, and/or error messages.
  - When the command outputs no messages or only informational and warning messages, the input data is valid. You can use the data with the `cc delete configuration data` command to update the configuration instance.
  - When the command outputs error messages, the input data is not valid. The `cc delete configuration data` command will fail if you use the data to attempt to update the configuration instance.

### Example When Executing on Command Central

The data to update a COMMON-PORTS configuration type instance is in the `c:\inputs\port_data.xml` file. To validate the data for the configuration instance with ID "COMMON-PORTS-com.softwareag.sshd.pid.properties" for the run-time component with ID "OSGI-SPM", which is installed in the installation with alias name "sag01" using the authorization of the user with user name "Administrator" and password "manage":

```
cc exec configuration validation sag01 OSGI-SPM update COMMON-PORTS-
com.softwareag.sshd.pid.properties --input c:\inputs\port_data.xml
--username Administrator --password manage
```

Because the `{--server | -s}` option is not specified, the command uses the default server. For more information, see ["--server | -s" on page 77](#).

### Example When Executing on Platform Manager

The data to update a COMMON-PORTS instance is in the `c:\inputs\port_data.xml` file. To validate the data for the configuration instance with ID "COMMON-PORTS-com.softwareag.sshd.pid.properties" for the run-time component that has the ID "OSGI-SPM" and is managed by the Platform Manager with host name "rubicon2" and port "8092":

```
cc exec configuration validation OSGI-SPM update COMMON-PORTS-
com.softwareag.sshd.pid.properties --input c:\inputs\port_data.xml
```

```
--server http://rubicon2:8092/spm --password secret
```

Because the `{--username | -u}` option is not specified, the command uses the default user name. For more information, see "[--username | -u](#)" on page 80. The command specifies "secret" for the user's password.

### Product-Specific Information

[Configuration Types that IntegrationServer-instanceName Supports](#)

[Configuration Types that My webMethods Server-ENGINE Supports](#)

### Related Commands

[cc create configuration data](#)

[cc exec configuration validation delete](#)

[cc exec configuration validation delete](#)

## Instance Management of Installed Products

---

### About Instance Management of Installed Products

Command Central allows you to create, update, and delete an instance of an installed product that supports multiple instances under the same installation directory. For example, you can create an instance of Integration Server with a different configuration or a different set of packages from an existing Integration Server installed in the same directory. You cannot change the name of an existing product instance, but you can update other configuration parameters provided when creating the instance. For example, you can update the list of recently installed packages for an Integration Server instance. You can also delete an installed product instance.

To manage instances of installed products, use the Command Central command line interface.

### Instance Management Commands

#### **cc create instances**

Creates a new instance of an installed product.

#### Syntax

- Command Central syntax:

```
cc create instances node_alias product
[key=value] | [-i file{.xml|.json|.properties}] [options]
```

- Platform Manager syntax:

```
cc create instances product
```

```
[key=value] | [-i file{.xml|.json|.properties}] [options]
```

## Arguments and Options

Argument or Option	Description
<code>node_alias</code>	<p>Command Central only.</p> <p>Required. Specifies the alias name of the installation in which to create the product instance.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<code>product</code>	<p>Required. Specifies the product ID of the installed product or run-time component for which you want to create a new instance.</p> <p>Valid values for this option are only the product IDs included in the list of products returned from the <code>cc list instances node_alias supportedproducts</code> command.</p>
<code>[key=value]</code>	<p>Optional. A list of properties that describe the elements of the new instance, such as name and port settings. The properties included in this list are product specific.</p>
<code>[-i [file{.xml .json .properties}]</code>	<p>Optional. Identifies an input file that contains the product instance data. For more information, see "<a href="#">--input   -i</a>" on <a href="#">page 70</a>.</p> <p>For the correct format of an XML properties file, see the Properties class in the Oracle Java Platform Standard Edition API specification.</p>
<code>[options]</code>	<p>Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see "<a href="#">Options for the Commands</a>" on <a href="#">page 59</a>.</p>

## Usage Notes

The command returns job information that you can monitor using the `cc list job manager` command.

## Example When Executing on Command Central

- To create the new instance for an installed Integration Server with instance name “is-instance2”, diagnostic port “8083”, JMX port “10058”, and primary port “8081” in the installation with alias name “productionNode2”:

```
cc create instances productionNode2 integrationServer
instance.name=is-instance2 diagnostic.port=8083
jmx.port=10058 primary.port=8081
```

## Examples When Executing on Platform Manager

- To create the new instance for an installed Integration Server with instance name “is-instance2”, diagnostic port “8083”, JMX port “10058”, and primary port “8081”:

```
cc create instances integrationServer instance.name=is-instance2
diagnostic.port=8083 jmx.port=10058 primary.port=8081
```

- To create the new instance for an installed Integration Server using the instance data in the `instance-settings.properties` file, located in the current directory:

```
cc create instances integrationServer -i instance-settings.properties
```

- To create the new instance for an installed Integration Server using the instance data in the `instance.settings.xml` file, located in the current directory:

```
cc create instances integrationServer -i instance-settings.xml
```

## Product-Specific Information

### Integration Server Instance Management

#### Related Commands

[cc delete instances](#)

[cc list instances supported products](#)

[cc list jobmanager jobs](#)

[cc update instances](#)

### cc delete instances

Deletes an existing instance of an installed product.

#### Syntax

- Command Central syntax:

```
cc delete instances node_alias componentid [options]
```

- Platform Manager syntax:

```
cc delete instances componentid [options]
```

```
options :  
[--force]
```

## Arguments and Options

Argument or Option	Description
<i>node_alias</i>	Command Central only.  Required. Specifies the alias name of the installation in which the run-time component is installed.  You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a> .
<i>componentid</i>	Required. Specifies the ID of the run-time component that you want to delete.
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands"</a> on page 59.

## Usage Notes

- The command returns job information that you can monitor using the `cc list job manager` command.
- You must stop an Integration Server instance before deleting the instance.

## Example When Executing on Command Central

To delete a run-time component with ID “integrationServer-default” that is installed in the installation with alias name “sag01”:

```
cc delete instances sag01 integrationServer-default
```

## Examples When Executing on Platform Manager

To delete a run-time component with ID “integrationServer-default”:

```
cc delete instances integrationServer-default
```

## Related Commands

[cc create instances](#)

[cc list instances supported products](#)

## cc list instances

Retrieves a list of the configuration properties of a specified run-time component.

### Syntax

- Command Central syntax:

```
cc list instances node_alias componentid [options]
```

- Platform Manager syntax:

```
cc list instances componentid [options]
```

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	<p>Command Central only.</p> <p>Required. Specifies the alias name of the installation in which the product is installed.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
<i>componentid</i>	<p>Required. Specifies the ID of a run-time component for which you want to retrieve configuration data.</p>
[ <i>options</i> ]	<p>Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a>.</p>

### Example When Executing on Command Central

To retrieve a list of configuration properties for a run-time component with ID “OSGI-IS” that runs in the installation with alias name “sag01”:

```
cc list instances sag01 OSGI-IS
```

### Example When Executing on Platform Manager

To retrieve a list of configuration properties for a run-time component with name “OSGI-IS”:

```
cc list instances OSGI-IS
```

### Related Commands

[cc create instances](#)

[cc delete instances](#)

[cc update instances](#)

## cc list instances supported products

Retrieves a list of products that support instance management.

### Syntax

- Command Central syntax:

```
cc list instances node_alias supportedproducts [options]
```

- Platform Manager syntax:

```
cc list instances supportedproducts [options]
```

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	<p>Command Central only.</p> <p>Required. Specifies the alias name of the installation in which the product is installed.</p> <p>You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a>.</p>
[ <i>options</i> ]	<p>Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see "<a href="#">Options for the Commands</a>" on page 59.</p>

### Example When Executing on Command Central

To retrieve a list of the products that support instance management in the installation with alias name "sag01":

```
cc list instances sag01 supportedproducts
```

### Example When Executing on Platform Manager

To retrieve a list of the products that support instance management in the installation:

```
cc list instances supportedproducts
```

### Related Commands

[cc create instances](#)

[cc update instances](#)

## cc update instances

Updates configuration properties of an existing instance of an installed product. For example, you might want to update a list of Integration Server packages.

### Syntax

- Command Central syntax:

```
cc update instances node_alias componentid
[key=value] | [-i file{.xml|.json|.properties}] [options]
```

- Platform Manager syntax:

```
cc update instances componentid
[key=value] | [-i file{.xml|.json|.properties}] [options]
```

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	Command Central only.  Required. Specifies the alias name of the installation in which the run-time component is installed.  You can view a list of installations and their aliases using <a href="#">cc list landscape nodes</a> .
<i>componentid</i>	Required. Specifies the ID of the run-time component for which you want to update configuration properties.
[ <i>key=value</i> ]	Optional. A list of properties that describe the elements of the new instance, such as name and port settings. The properties included in this list are product specific.
[-i [ <i>file</i> {.xml .json .properties}]]	Optional. Identifies an input file that contains the new configuration data for the run-time component. For more information, see " <a href="#">--input   -i</a> " on page 70.  For the correct format of an XML properties file, see the Properties class in the Oracle Java Platform Standard Edition API specification.

Argument or Option	Description
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Example When Executing on Command Central

To update the "WmBusinessRules" package on an Integration Server with ID "integrationServer-default" that is installed in the installation with alias name "sag01":

```
cc update instances sag01 integrationServer-default
package.list=WmBusinessRules
```

### Examples When Executing on Platform Manager

- To update the "WmBusinessRules" package on an Integration Server with ID "integrationServer-default":

```
cc update instances integrationServer-default
package.list=WmBusinessRules
```

- To update configuration properties for an installed run-time component with ID "OSGI-IS\_default" using configuration data in the instance-settings.properties file, located in the current directory:

```
cc update instances OSGI-IS_default -i instance-settings.properties
```

- To update configuration properties for an installed run-time component with ID "OSGI-IS\_default" using configuration data in the instance-settings.xml file, located in the current directory:

```
cc update instances OSGI-IS_default -i instance-settings.xml
```

### Related Commands

[cc create instances](#)

[cc delete instances](#)

[cc list instances supported products](#)

---

# 11 Repository Management

---

■ Creating an Image Repository .....	286
■ Registering a Master Repository .....	287
■ Deleting a Repository .....	288
■ Editing Image Repository Details .....	288
■ Editing Master Repository Details .....	289
■ Repositories Commands .....	289

## Creating an Image Repository

Use the following procedure to create an image repository on Command Central. You can point to an image repository created on the Command Central server when applying a template.

### To create an image repository

1. In Command Central, go to **Views > Repositories**.
2. Click  and select **Image**.
3. In the Create image repository dialog box, specify:

Field	Description
<b>Repository name</b>	Required. A unique name for the repository. The only valid characters in a repository name are ASCII characters, numbers, underscore (_), dot (.), and a hyphen (-).
<b>Repository description</b>	Optional. A description of the repository. This description is displayed in the repositories list on the Repositories page.
<b>Repository contents</b>	Select whether the repository contains products or fixes.
<b>Image file on server</b>	Required. The fully qualified path to an existing image file on the Command Central server to which the repository refers.  If you do not specify a path, you must upload an image file using the <b>Upload image file</b> field. Command Central populates the <b>Image file on server</b> field with the path to the uploaded image file.
<b>Upload image file</b>	Select and upload an existing image file (created with Installer for products or with the Update Manager for fixes) from the file system to the Command Central server.

4. Click **OK**.

Command Central adds the new image repository in the repositories list.

In the repositories list, you can view repository details, such as name, description, contents, and type.

## Registering a Master Repository

Use the following procedure to register a master repository on Command Central. You can point to a master repository registered with Command Central when applying a template.

### To register a master repository

1. In Command Central, go to **Views > Repositories**.
2. Click  and select **Master**.
3. In the Register master repository dialog box, specify values for the following fields:

Field	Description
<b>Repository name</b>	Required. A unique name for the repository. The only valid characters in a repository name are ASCII characters, numbers, underscore (_), dot (.), and a hyphen (-).
<b>Repository description</b>	Optional. A description of the repository. This description is displayed in the repositories list on the Repositories page.
<b>Repository contents</b>	Select whether the repository contains products or fixes.
<b>Credentials</b>	Required. Enter valid user name and password for the Empower website.
<b>Version</b>	Required. Select a product repository version. The version for a fix repository is populated automatically.
<b>Advanced</b>	Optional. The URL of the server from which to install products.
	<b>Important</b> Use this field only if directed by Software AG Global Support.

4. Click **OK**.

Command Central adds the new master repository in the repositories list.

In the repositories list, you can view repository details, such as name, description, contents, and type.

## Deleting a Repository

Use the following procedure to delete a repository registered with Command Central.

### To delete a repository

1. In Command Central, go to **Views > Repositories**.
2. From the repositories list, select the repository to delete.
3. Click .
4. Click **OK**.

When deleting an image repository, Command Central deletes both the repository and the image file that the repository refers to.

## Editing Image Repository Details

You can edit the description for an image repository and update the image file to which the repository refers.

### To edit a registered image repository

1. In Command Central, go to **Views > Repositories**.
2. Click the name of the repository that you want to edit.
3. In the Edit image repository dialog box, you can edit *only* the values in the following fields:

Use this field...	To...
<b>Repository description</b>	Replace the repository description.
<b>Image file on server</b>	Change the fully qualified path to an existing image file on the Command Central server which the repository uses by reference.  If you do not specify a path, you must upload an image file using the <b>Upload image file</b> field. Command Central populates the <b>Image file on server</b> field with the path to the uploaded image file.
<b>Upload image file</b>	Select and upload a new image file that replaces the image file you uploaded when adding the repository.

4. Click **OK**.

## Editing Master Repository Details

You can edit the description for a master repository and change your repository credentials.

### To edit a registered master image repository

1. In Command Central, go to **Views > Repositories**.
2. Click the name of the repository that you want to edit.
3. In the Edit master repository dialog box, you can edit *only* the values in the following fields:

Use this field...	To...
<b>Repository description</b>	Replace the repository description.
<b>Credentials</b>	Change the user name and password for the registered master repository.
<b>Version</b>	Change the product or fix version.
<b>Advanced</b>	Specify a different server URL from which to download products or fixes.

4. Click **OK**.

## Repositories Commands

### cc add repository fixes

DEPRECATED. Use [cc add repository](#). Registers a fix repository.

#### Syntax

- Command Central syntax:

```
cc add repository fixes node_alias path=path [options]
```

- Platform Manager syntax:

```
cc add repository fixes path=path [options]
```

## Arguments and Options

Argument or Option	Description
<code>node_alias</code>	Required. Only for Command Central. Specifies the alias name of the Platform Manager to which you want to add the fix repository.
<code>path=path</code>	<p>Required. The path to the fix repository to register.</p> <ul style="list-style-type: none"> <li>■ If the fix repository is remote, specify the URI with http or https.</li> <li>■ If the fix repository is local, the path may refer either to an image archive created with Software AG Update Manager (containing images or products), or to a local directory. When the location is a directory: <ul style="list-style-type: none"> <li>■ If the directory that does not exist or is empty, Platform Manager creates the directory and populates it with an empty p2 repository.</li> <li>■ If the directory exists and is not empty, ensure that it is a valid p2 repository of products.</li> </ul> </li> </ul>
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

## Usage Notes

After adding a fix repository by using the `discover` or `register` command, Platform Manager creates a new configuration instance that you use to configure the repository. For example, you can add access credentials to a remote repository by using the new configuration instance for a registered fix repository as follows:

1. Execute the following command:

```
cc get configuration data node-spm OSGI-SPM-ENGINE
REPOSITORY-empower-fix-image.zip
```

2. Save the result to a file with name "repoCredentials.prop".
3. In a text editor, edit the `repoCredentials.prop` file as follows:

```
user=EmpowerUserEmailAddress
password=EmpowerUserPassword
```

#### 4. Execute the following commands:

```
cc update configuration data node-spm
OSGI-SPM-ENGINE REPOSITORY-empower-fix-image.zip
-i repoCredentials.prop
cc get configuration data node-spm
OSGI-SPM-ENGINE REPOSITORY-empower-fix-image.zip
```

The result contains the updated password and username.

#### Example When Executing on Command Central

To add an image file with name “image.zip” as a repository in the Platform Manager registered as “is-dev”:

```
cc add repository fixes is-dev path=file:///home/neta/work/image.zip
```

If you do not specify a value for `CC_SERVER`, the request is sent to the Command Central server on “localhost:8090”.

#### Example When Executing on Platform Manager

To add an image file with name “image.zip” as a repository in the Platform Manager hosted on a server with host name “rubicon” and port “8092”:

```
cc add repository fixes path=file:///home/neta/work/image.zip
--server http://rubicon:8092/spm
```

#### Related Commands

[cc exec repository discover](#)

[cc exec repository register](#)

[cc list repository](#)

[cc list repository discover](#)

## cc add repository products

DEPRECATED. Use [cc add repository](#). Registers product repositories.

#### Syntax

##### ■ Command Central syntax:

```
cc add repository products node_alias path=path [options]
```

##### ■ Platform Manager syntax:

```
cc add repository products path=path [options]
```

## Arguments and Options

Argument or Option	Description
<code>node_alias</code>	Required. Only for Command Central. Specifies the alias name of the Platform Manager to which you want to add the product repository.
<code>path=path</code>	<p>Required. The path to the product repository to register.</p> <ul style="list-style-type: none"> <li>■ If the product repository is remote, specify the URI with <code>http</code> or <code>https</code>.</li> <li>■ If the product repository is local, the path may refer either to an image archive created with Software AG Installer (containing images or products), or to a local directory. When the location is a directory: <ul style="list-style-type: none"> <li>■ If the directory that does not exist or is empty, Platform Manager creates the directory and populates it with an empty p2 repository.</li> <li>■ If the directory exists and is not empty, ensure that it is a valid p2 repository of products.</li> </ul> </li> </ul>
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

## Usage Notes

After adding a product repository by using the `discover` or `register` command, Platform Manager creates a new configuration instance that you use to configure the repository. For example, you can add access credentials to a remote repository by using the new configuration instance for a registered product repository as follows:

1. To select the registered product repository for which you want to configure credentials:
  - Execute the command:

```
cc list repository products node-spm
```
  - Select the name of the repository from the list.

- Execute the following command:

```
cc get configuration data node-spm
OSGI-SPM-ENGINE REPOSITORY-sdc.softwareag.com-cgi
-bin-dataserwebM95.cgi-image.zip
```

- Save the result to a file with name “repoCredentials.prop”.
- In a text editor, edit the repoCredentials.prop file as follows:

```
user=EmpowerUserEmailAddress
password=EmpowerUserPassword
```

- Execute the following commands:

```
cc update configuration data node-spm OSGI-SPM-ENGINE
REPOSITORY-sdc.softwareag.com-cgi-bin-dataserwebM95.cgi
-image.zip -i repoCredentials.prop
cc get configuration data node-spm OSGI-SPM-ENGINE
REPOSITORY-sdc.softwareag.com-cgi-bin-dataserwebM95.cgi-image.zip
```

The result contains the updated password and username.

### Example When Executing on Command Central

To add an existing image file with name “image.zip” (created using the Software AG Installer) as a repository in the Platform Manager registered as “is-dev”:

```
cc add repository products is-dev path=file:///home/neta/work/image.zip
```

If you do not specify a value for `CC_SERVER`, the request is sent to the Command Central server on “localhost:8090”.

### Example When Executing on Platform Manager

To add an existing image file with name “image.zip” (created using the Software AG Installer) as a repository in the Platform Manager hosted on a server with host name “rubicon” and port “8092”:

```
cc add repository products
path=file:///home/neta/work/image.zip -server http://rubicon:8092/spm
```

### Related Commands

[cc exec repository discover](#)

[cc exec repository register](#)

[cc list repository](#)

[cc list repository discover](#)

## cc add repository

Adds a product or fix repository.

### Syntax

- Command Central syntax:

- Using arguments:

```
cc add repository {products | fixes} name=repo_name [type=type]
location=URL [description="description"] [options]
```

- Using an input data file:

```
cc add repository {products | fixes} name=repo_name
--input | i filename.zip [description="description"] [options]
```

- Not supported by Platform Manager.

## Arguments and Options

Argument or Option	Description
<code>name=repo_name</code>	Required. The name of the product or fix repository to add.
<code>[type=type]</code>	<p>Optional. The type of the product or fix repository to add. Valid values are:</p> <ul style="list-style-type: none"> <li>■ image</li> <li>■ mirror</li> <li>■ master</li> </ul> <p>When adding an image repository, if the location URL starts with "file://", this parameter is optional.</p>
<code>location=URL</code>	<p>Required. A valid URL that points to the location where the repository is added. If <code>location</code> points to an installation image file, the image file must exist on the Command Central server. If the image file does not exist, the repository is not created.</p> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p><b>Important:</b> No repositories can point to the same location.</p> </div>
<code>--input   -i filename.zip</code>	<p>Required. Identifies an image archive file created with Software AG Installer or Software AG Update Manager (containing products or fixes). For more information about the <code>--input   i</code> option, see "<a href="#">--input   -i</a>" on page 70.</p>
<code>[description="description"]</code>	Optional. A description for the repository.

Argument or Option	Description
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Examples When Executing on Command Central

- To add a repository with name "test" at the "http://test/repo" location:

```
cc add repository products name=test location=http://test/repo
cc add repository fixes name=test location=http://test/repo
```
- To upload an image file named "image.zip" from the current directory to Command Central and create a repository with name "test" that points to that image:

```
cc add repository products name=test -i image.zip
cc add repository fixes name=test -i image.zip
```

### Related Commands

[cc exec repository register](#)

[cc exec repository discover](#)

[cc list repository](#)

[cc update repository](#)

## cc delete repository

Deletes a registered product or fix repository.

### Syntax

- Command Central syntax:

```
cc delete repository {products | fixes} repo_name
[deleteImage={true | false}] [options]
```

*options:*  
[`--force`]
- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
<code>repo_name</code>	Required. Specifies the name of the repository that you want to delete.

Argument or Option	Description
<code>deleteImage={true   false}</code>	<p>Optional. Whether to delete the image file referenced by the repository from the file system. Valid values:</p> <ul style="list-style-type: none"> <li>■ <code>true</code> Delete the image file.</li> <li>■ <code>false</code> (default) Do not delete the image file.</li> </ul> <p>When you do not include this argument, Command Central does not delete the image file from the file system.</p>
<code>[options]</code>	<p>Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a>.</p>

### Examples When Executing on Command Central

- To delete a repository with name “repo1” including the image file that the repository references:

```
cc delete repository products REPOSITORY-repo1 deleteImage=true
cc delete repository fixes REPOSITORY-repo1 deleteImage=true
```

- To delete a product repository with name “test” without deleting the image file that the repository refers to:

```
cc delete repository products test
```

### Related Commands

[cc add repository](#)

[cc exec repository discover](#)

[cc list repository](#)

[cc update repository](#)

## cc delete repository with *node\_alias*

DEPRECATED. Use [cc delete repository](#). Deletes a registered product or fix repository on a specified Platform Manager node. The command removes access to the repository without deleting actual repository data.

### Syntax

- Command Central syntax:

```
cc delete repository {products | fixes} node_alias repo_name [options]
```

```
options :
[--force]
```

- Platform Manager syntax:

```
cc delete repository {products | fixes} node_alias repo_name [options]
```

### Arguments and Options

Argument or Option	Description
<code>node_alias</code>	Required. Only for Command Central. Specifies the alias name of the Platform Manager from which you want to delete the repository.
<code>repo_name</code>	Required. Specifies the name of the repository that you want to delete.
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Examples When Executing on Command Central

- To delete a repository with name “repo1” from the Platform Manager registered as “is-dev”:

```
cc delete repository products is-dev REPOSITORY-repo1
cc delete repository fixes is-dev REPOSITORY-repo1
```

### Related Commands

[cc add repository](#)

[cc exec repository discover](#)

[cc list repository](#)

[cc update repository](#)

## cc delete repositories

Deletes all registered product or fix repositories.

### Syntax

- Command Central syntax:

```
cc delete repository {products | fixes}[options]
```

```
options :
[--force]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

To prevent loss of information, you are prompted to confirm if you want to delete all repositories of the specified type. Use the `--force` command option to override the confirmation request. For more information, see ["--force" on page 66](#).

### Examples When Executing on Command Central

- To delete all registered product repositories after user confirmation:
 

```
cc delete repository products
```
- To delete all registered fix repositories without user confirmation:
 

```
cc delete repository products --force
```

### Related Commands

[cc delete repository](#)

[cc delete repository with node\\_alias](#)

## cc exec repository discover

Finds product and fix repositories for the specified host, name, and port and adds the discovered repositories to Command Central.

### Syntax

- Command Central syntax:
 

```
cc exec repository {products | fixes} discover [host=host]
[name=repo_name] [port=port] [options]
```
- Platform Manager syntax:
 

```
cc exec repository discover [host=host] name=repo_name
[port=port] [options]
```

## Arguments and Options

Argument or Option	Description
[ <code>host=host</code> ]	Optional. The host name or IP address of the system hosting the repositories. If you do not specify a value, Command Central goes to the Empower website.
[ <code>name=repo_name</code> ]	Optional. The name of the repository on the specified host machine. If you do not specify a value, Command Central lists all repositories on the host machine.
[ <code>port=port</code> ]	Optional. The port number of the specified host machine.
[ <code>options</code> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Example When Executing on Command Central and Platform Manager

To add a product repository with name "SuiteProd" from a server with host name "sag":

```
cc exec repository products discover host=sag name=SuiteProd
cc exec repository fixes discover host=sag name=SuiteProd
```

If you do not specify a value for `CC_SERVER`, the request is sent to the Command Central server on "localhost:8090".

### Related Commands

[cc add repository](#)

[cc exec repository register](#)

[cc list repository](#)

[cc list repository discover](#)

[cc update repository](#)

### **cc exec repository discover with *node\_alias***

DEPRECATED. Use [cc exec repository discover](#). Discovers public product or fix repositories.

## Syntax

### ■ Command Central syntax:

```
cc exec repository {products | fixes} node_alias discover
[host=host] [name=repo_name] [port=port] [options]
```

### ■ Platform Manager syntax:

```
cc exec repository {products | fixes} discover [host=host]
[name=repo_name] [port=port] [options]
```

## Arguments and Options

Argument or Option	Description
<code>node_alias</code>	Required. Only for Command Central. Specifies the alias name of the Platform Manager for which you want discover repositories.
<code>[host=host]</code>	Optional. The host name or IP address of the system hosting the repositories. If you do not specify a value, Command Central and Platform Manager go to the Empower website.
<code>[name=repo_name]</code>	Required when you specify a value for <code>host</code> . The name of the repository on the specified host machine. If you do not specify a value, Command Central lists all repositories on the host machine.
<code>[port=port]</code>	Optional. The port number of the specified host machine.
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

## Usage Notes

After adding a repository by using the `discover` or `register` command, Platform Manager creates a new configuration instance that you use to configure the repository. For an example how to add access credentials to a remote repository, see the usage notes section in:

- For product repositories, ["cc add repository products" on page 291](#)
- For fix repositories, ["cc add repository fixes" on page 289](#)

### Example When Executing on Command Central

To add a product repository with name “SuiteProd” from a server with host name “sag” to the Platform Manager registered as “is-dev”:

```
cc exec repository products is-dev discover host=sag name=SuiteProd
```

If you do not specify a value for `CC_SERVER`, the request is sent to the Command Central server on “localhost:8090”.

### Example When Executing on Platform Manager

To add a fix repository with name “QARepo” from a server with host name “sag” to the Platform Manager hosted on a server with host name “rubicon” and port “8092”:

```
cc exec repository fixes discover host=sag name=QARepo -s rubicon:8092/spm
```

If you do not specify a value for `CC_SERVER`, the request is sent to the Command Central server on “localhost:8090”.

### Related Commands

[cc add repository products](#)

[cc list repository](#)

[cc list repository discover](#)

## cc exec repository register

Copies a product or fix repository, including its image file, to a new Platform Manager node.

### Syntax

- Command Central syntax:

```
cc exec repository {products | fixes} register node_alias repo_name
[options]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	Required. Specifies the alias name of the Platform Manager to which you want to copy the repository.
<i>repo_name</i>	Required. Specified the name of the repository that you want to copy.

Argument or Option	Description
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Examples When Executing on Command Central

- To copy the repository with name “repo1” to the Platform Manager with name “node1”:

```
cc exec repository products register node1 repo1
cc exec repository fixes register node1 repo1
```

- To copy the repository with name “QARepo” to the local Platform Manager installation:

```
cc exec repository fixes register local QARepo
```

### Related Commands

[cc add repository](#)

[cc delete repository](#)

[cc exec repository discover](#)

[cc list repository](#)

[cc update repository](#)

## cc exec repository fixes export

Generates a fix image archive from a fix repository.

### Syntax

- Command Central syntax:

```
cc exec repository fixes export repo_Name
dest=filename.zip artifacts=fixName1[_version],fixName2[_version]
[options]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
<code>repo_name</code>	Required. Specifies the name of the repository from which you want to generate a fix image archive.

Argument or Option	Description
<code>dest=filename.zip</code>	Required. A name for the generated fix image archive.
<code>artifacts=fixName1 [_version], fixName2 [_version]</code>	Required. The names of the fixes to include in the image file. Specifying the fix version is optional.
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Example When Executing on Command Central

To generate a fix image archive with name “myfixes.zip” that contains a fix with name “wMFix.SPM.TEST” from the Empower repository:

```
cc exec repository fixes export Empower dest=myfixes.zip
artifacts=wMFix.SPM.TEST
```

### Related Commands

[cc add repository](#)

[cc delete repository](#)

[cc list repository](#)

## cc list repository

Lists registered product or fix repositories.

### Syntax

- Command Central syntax:  
`cc list repository {products | fixes} [options]`
- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For

Argument or Option	Description
	a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Examples When Executing on Command Central

To list the registered repositories on a Command Central server with host name "rubicon" and port "8490":

```
cc list repository products -server http://rubicon:8490/cce
cc list repository fixes -server http://rubicon:8490/cce
```

If you do not specify a value for `CC_SERVER`, the request is sent to the Command Central server on "localhost:8090".

### Related Commands

[cc add repository](#)

[cc list repository discover](#)

[cc update repository](#)

## cc list repository with *node\_alias*

DEPRECATED. Use [cc list repository](#). Lists registered product and fix repositories on a specified Platform Manager node.

### Syntax

- Command Central syntax:

```
cc list repository {products | fixes} node_alias
[refresh=value] [options]
```

- Platform Manager syntax:

```
cc list repository {products | fixes} [refresh=value] [options]
```

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	Required. Command Central only. Specifies the alias name of a Platform Manager installation. You can determine installation alias names using the <a href="#">cc list landscape nodes</a> command.
<i>refresh=value</i>	Optional. Whether to send a new request to the specified Platform Manager to update the

Argument or Option	Description
	<p>Command Central server cache with data about the registered repositories.</p> <ul style="list-style-type: none"> <li>■ <code>true</code> Sends a request.</li> <li>■ <code>false</code> (default) Does not send a request.</li> </ul>
<code>[options]</code>	<p>Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a>.</p>

### Usage Notes

To list the contents of a fix repository, Command Central must connect to a local or remote Platform Manager that is running and has access to the fix repository.

### Example When Executing on Command Central

- To list the registered product repositories for a Platform Manager registered as “is-dev” on a Command Central server with host name “rubicon” and port “8490”:

```
cc list repository products is-dev -server http://rubicon:8490/cce
```

- To update the Command Central server cache and list all registered fix repositories for a Platform Manager registered as “is-dev”:

```
cc list repository fixes is-dev refresh=true
```

If you do not specify a value for `CC_SERVER`, the request is sent to the Command Central server on “localhost:8090”.

### Example When Executing on Platform Manager

To list the registered product repositories for a Platform Manager hosted on a server with host name “rubicon” and port “8092”:

```
cc list repository products -server http://rubicon:8092/spm
```

### Related Commands

[cc list repository](#)

[cc list repository discover](#)

## cc list repository discover

Finds product or fix repositories for the specified host, name, and port, but does not add the discovered repositories to Command Central.

**Syntax**

- Command Central syntax:

```
cc list repository discover [host=host] [name=repo_name]
[port=port] [options]
```

- Not supported by Platform Manager.

**Arguments and Options**

Argument or Option	Description
[ <i>host=host</i> ]	Optional. The host name or IP address of the system hosting the repositories. If you do not specify a value, Command Central goes to the Empower website.
[ <i>name=repo_name</i> ]	Optional. The name of the repository on the specified host machine. If you do not specify a value, Command Central lists all repositories on the host machine.
[ <i>port=port</i> ]	Optional. The port number of the specified host machine.
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

**Example When Executing on Command Central**

To find a product repository with name "SuiteProd" on a server with host name "sag":

```
cc list repository discover host=sag name=SuiteProd
```

If you do not specify a value for `CC_SERVER`, the request is sent to the Command Central server on "localhost:8090".

**Related Commands**

[cc add repository](#)

[cc exec repository discover](#)

[cc exec repository register](#)

[cc list repository](#)

[cc update repository](#)

## cc list repository fixes content

Lists the fixes available in a fix repository.

### Syntax

- Command Central syntax:

```
cc list repository fixes content repo_name
[products=productId1_[version],productId2_[version] | INSTALLED]
platform=OS [nodeAlias=nodeAlias] [options]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>repo_name</i>	Required. The name of the fix repository for which you want to list available fixes.
[products= <i>productId1_[version],productId2_[version]   INSTALLED</i> ]	Optional. The IDs and version of the products for which to list available fixes. Use commas to separate each product ID.  You can determine the IDs for run-time components using the <a href="#">cc list inventory components</a> command.  When you specify <code>INSTALLED</code> Command Central lists the fixes that match the operating system and installed products on the target installation.
[platform= <i>OS</i> ]	Optional. The ID of the operating system for which to list fixes.
[nodeAlias= <i>nodeAlias</i> ]	Required. Specifies the alias name of a Platform Manager installation. You can determine installation alias names using the <a href="#">cc list landscape nodes</a> command.
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of

Argument or Option	Description
	the options, see <a href="#">"Options for the Commands" on page 59.</a>

### Examples When Executing on Command Central

- To list all available fixes for all products and versions in a fix repository with name "repo1":
 

```
cc list repository fixes content repo1
```
- To list the fixes available for products with IDs "SPM" and "OSGI" for all versions in a repository with name "repo1":
 

```
cc list repository fixes content repo1 products=SPM,OSGI
```
- To list the fixes available for products with IDs "SPM" and "OSGI", and version "9.8" for the operating system with ID "W64" in a fix repository with name "repo1":
 

```
cc list repository fixes content repo1 platform=W64
products=SPM_9.8,OSGI_9.8
```
- To list fixes that match the operating system and installed products on a target node with name "sag01", available in a fix repository with name "repo1":
 

```
cc list repository fixes content repo1 products=INSTALLED
nodeAlias=sag01
```

### Related Commands

[cc list repository](#)

[cc list repository fixes dependencies](#)

[cc list repository fixes readme](#)

[cc list repository platforms](#)

[cc list inventory components](#)

[cc list landscape nodes](#)

## cc list repository fixes dependencies

Checks the dependencies for a fix.

### Syntax

- Command Central syntax:
 

```
cc list repository fixes dependencies repo_name fix_name [_version] [options]
```
- Not supported by Platform Manager.

## Arguments and Options

Argument or Option	Description
<code>repo_name</code>	Required. The name of the fix repository that contains the fix.
<code>fix_name[_version]</code>	Required. The name of the fix for which you want to check dependencies. Specifying the fix version is optional.
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Example When Executing on Command Central

To check the dependencies for a fix with name "wMFix.SPM.TEST" in a fix repository with name "repo1":

```
cc list repository fixes dependencies repo1 wMFix.SPM.TEST
```

### Related Commands

[cc list repository](#)

[cc list repository fixes content](#)

[cc list repository fixes readme](#)

[cc list repository platforms](#)

## cc list repository fixes readme

Retrieves the readme for a fix.

### Syntax

- Command Central syntax:

```
cc list repository fixes readme repo_name fix_name[_version] [options]
```

- Not supported by Platform Manager.

## Arguments and Options

Argument or Option	Description
<code>repo_name</code>	Required. The name of the fix repository that contains the fix.
<code>fix_name[_version]</code>	Required. The name of the fix for which you want to retrieve the readme. Specifying the fix version is optional.
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Example When Executing on Command Central

To retrieve the readme for a fix with name “wMFix.SPM.TEST” in a fix repository with name “repo1”:

```
cc list repository fixes readme repo1 wMFix.SPM.TEST
```

### Related Commands

[cc list repository](#)

[cc list repository fixes content](#)

[cc list repository fixes dependencies](#)

[cc list repository platforms](#)

## cc list repository platforms

Lists all supported operating systems on the specified Platform Manager node.

### Syntax

- Command Central syntax:

```
cc list repository platforms node_alias [options]
```

- Platform Manager syntax:

```
cc list repository platforms [options]
```

## Arguments and Options

Argument or Option	Description
<code>node_alias</code>	Specifies the alias name of the Platform Manager node for which you want to list operating systems available for bootstrapping or provisioning.
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Example When Executing on Command Central

To list all operating systems available on the local Platform Manager node on which Command Central can execute repository and provisioning operations:

```
cc list repository platforms local
```

### Example When Executing on Platform Manager

To list all operating systems available on the local Platform Manager node on which Command Central can execute repository and provisioning operations:

```
cc list repository platforms
```

### Related Commands

[cc list repository](#)

[cc list repository discover](#)

[cc list repository fixes content](#)

[cc list repository fixes dependencies](#)

## cc list repository products content

Lists the products available in a product repository.

### Syntax

- Command Central syntax:

```
cc list repository products content repo_name
[products=productId_[version],productId2_[version | INSTALLED]
[nodeAlias=nodeAlias] [options]
```

- Not supported by Platform Manager.

## Arguments and Options

Argument or Option	Description
<code>repo_name</code>	Required. The name of the product repository for which you want to list available products.
<code>[products= productid1_[version], productid2 _[version]   INSTALLED]</code>	<p>Optional. The IDs and version of the products to list. Use commas to separate each product ID.</p> <p>You can determine the IDs for run-time components using the <a href="#">cc list inventory components</a> command.</p> <p>When you specify <code>INSTALLED</code> Command Central lists the products that match the operating system and the products installed on the target installation.</p>
<code>[nodeAlias=nodeAlias]</code>	Required. Specifies the alias name of a Platform Manager installation. You can determine installation alias names using the <a href="#">cc list landscape nodes</a> command.
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

## Examples When Executing on Command Central

- To list all available products for all versions in a product repository with name "repo1":
 

```
cc list repository products content repo1
```
- To list the products with IDs "SPM" and "OSGI", and version "9.8" in a fix repository with name "repo1":
 

```
cc list repository products content repo1  
products=SPM_9.8,OSGI_9.8
```
- To list products that match the operating system and installed products on a target node with name "sag01", available in a product repository with name "repo1":
 

```
cc list repository products content repo1 products=INSTALLED  
nodeAlias=sag01
```

## Related Commands

[cc list repository](#)

[cc list repository products dependencies](#)

[cc list inventory components](#)

[cc list landscape nodes](#)

## cc list repository products dependencies

Checks the dependencies for a product.

**Important:** Image repositories are not supported by the command.

### Syntax

- Command Central syntax:

```
cc list repository products dependencies repo_name productId[_version] [options]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>repo_name</i>	Required. The name of a master product repository that contains the product.
<i>productId[_version]</i>	Required. The ID of the product for which you want to check dependencies. Specifying the product version is optional.
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Example When Executing on Command Central

To check the dependencies for a product with ID "IS" in a fix repository with name "repo1":

```
cc list repository products dependencies repo1 IS
```

### Related Commands

[cc list repository](#)

[cc list repository products content](#)

## cc update repository

Updates a repository using data from an XML file. The XML file must contain a valid XML representation of the repository and the name of the repository.

### Syntax

- Command Central syntax:

```
cc update repository {products | fixes} -i filename.xml [options]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
<code>-i filename.xml</code>	Required. Specified the name of the XML file that contains the repository data. For more information, see " <a href="#">--input   -i</a> " on page 70.
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Example When Executing on Command Central

To update a repository using data from the "repository.xml" file, located in the current directory:

```
cc update repository products -i repository.xml
cc update repository fixes -i repository.xml
```

### Related Commands

[cc add repository](#)

[cc exec repository discover](#)

[cc exec repository register](#)

[cc list repository](#)

[cc update repository details](#)

## cc update repository details

Updates repository details, such as user credentials, location, and description.

### Syntax

- Command Central syntax:

```
cc update repository {products | fixes} repo_name [username=username]
[password=password] [location=image_filename | URL]
[description="description"] [options]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>repo_name</i>	Required. Specifies the name of the repository that you want to update.
[username= <i>username</i> ]	Optional. Specify a valid user name for the product or fix repository.
[password= <i>password</i> ]	Optional. Specify a valid password to the product or fix repository.
[location= <i>image_file</i>   <i>URL</i> ]	Optional. The name of the image file or valid URL to the new location of the image file referenced by the repository.
[description=" <i>description</i> "]	Optional. The new description of the repository.
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Example When Executing on Command Central

- To update the user name and password for a fix repository with name "QARepo":

```
cc update repository fixes QARepo username=sum password=secret
```

- To change the location of the file “image.zip”referenced by the repository with name “repo1”:

```
cc update repository products repo1 location=file:///vmtest/image.zip  
cc update repository fixes repo1 location=file:///vmtest/image.zip
```

To change the location of the file “image.zip”referenced by the repository with name “repo1”:

```
cc update repository products repo1 location=file:///vmtest/image.zip  
cc update repository fixes repo1 location=file:///vmtest/image.zip
```

### Related Commands

[cc add repository](#)

[cc exec repository discover](#)

[cc exec repository register](#)

[cc list repository](#)

[cc update repository](#)

---

# 12

## Template-based Provisioning

---

■ About Template-based Provisioning .....	318
■ Provisioning Installations .....	318
■ Provisioning Environments .....	327
■ Template Commands .....	338

---

## About Template-based Provisioning

---

With Command Central, you can use templates to install, update, and configure the webMethods landscape. Command Central provides and supports two types of templates:

- Snapshot Templates
- Composite Templates

---

## Provisioning Installations

---

You define a snapshot template by pointing to an existing managed installation and selecting the products, fixes, and configuration that you want to include in the snapshot template. Based on the intended use for the snapshot template you define, a snapshot template may contain any combination of products, fixes, and configuration available on the source installation. Command Central creates and stores the new snapshot template in the Command Central file system. The snapshot template contains all the required information about the installed products, the applied fixes, and the available configuration on the source installation. However, a snapshot template can contain only fixes or only configuration. For example, when you want to patch a number of existing installations, you can define a snapshot template that contains only the fixes you want to apply to each installation. If required, you can use a text editor to modify the snapshot template files.

After you create a snapshot template from a source instance, you can either apply the snapshot template immediately to clone the source installation, or store the snapshot template to create one or more instances at a later time. You can apply the snapshot template on a target machine that hosts only Platform Manager. When applying the snapshot template, Command Central installs the products and applies the fixes and configuration included in the snapshot template on the target machine.

**Important:** When applying a snapshot template with configuration that relies on absolute paths, ensure that the target nodes run on the same operating system and have the same root installation directory.

When applying a stored snapshot template, you select which parts of the template you want to apply. As a result, the set of products, fixes and configuration on the target machine may be different from the set on the source installation. The following table lists some examples of situations when applying only one of the elements available in a snapshot template is required:

**Important:** To avoid problems with shared system resources, apply only one snapshot template with products at a time on the machine. Make sure the entire product template application is complete before applying any other product snapshot template on the machine.

Apply only...	When...
Products	A different snapshot template is used to apply fixes or configuration.
Fixes	The products are already installed on the target machine and you need to apply fixes. In this scenario, you can test the fixes you want to apply in a test environment, create a fix-only snapshot template, apply the snapshot template to the staging environment where you test the fixes thoroughly, and finally apply the template to the production environment.
Configuration	Snapshot templates capture different product instances. For example, use one configuration snapshot template for a Platform Manager configuration, a second snapshot template for an Integration Server configuration (default instance), and a third snapshot template to configure Universal Messaging (default instance).

When a template configuration is applied in replace mode, any local configuration changes on the target machine get rolled back to ensures compliance between the configuration on the source and target instances. To preserve the local configuration settings on the target machine, you must apply a template configuration in merge mode.

**Important**The configuration included in a snapshot template is limited to the configuration supported by the Command Central product plug-in.

To install the products and apply the fixes from the snapshot template, Command Central accesses a public product or fix repository that is registered with Command Central. For example, to install products from the Empower website, you must register the Empower website as a product repository with Command Central. When applying a snapshot template that contains products, Command Central will access the Empower website to install the relevant products. You can register two types of repositories with Command Central:

- **Master repository:** a remote product repository, for example the Empower website
- **Image repository:** an image file that contains products or fixes

To manage snapshot templates and repositories, you can use either the Command Central web user interface or the command line interface.

## Creating a Snapshot Template

With Command Central you create a new snapshot template by pointing to an existing managed installation from which you include the products, fixes, and configuration you want to provision.

### To create a snapshot template from a managed installation

1. In the Environments pane, select the environment that contains the installation you want.
2. Select the **Installations** tab.
3. On the **Installations** tab, click the installation from which you want to create a snapshot template.
4. Click  and select **Save as template**.
5. In the Save Template dialog box, provide the following information:

Field...	Specify...
<b>Alias</b>	A unique name for the snapshot template. The only valid characters in a snapshot template name are ASCII characters, numbers, underscore (_), dot (.), and a hyphen (-).
<b>Description</b>	Optional. A description of the snapshot template. This description is displayed in the template list on the Template page of the Apply Template wizard.
<b>Include Products</b>	Whether to include products in the snapshot template. Select the check box to include products (default).
<b>Include Fixes</b>	Whether to include fixes in the snapshot template. Select the check box to include fixes (default).
<b>Include Files</b>	Whether to include files in the snapshot template. Select the check box to include files (default).
<b>Include Configuration</b>	Whether to include configuration in the snapshot template. Select the check box to include product configuration (default). When you include configuration, you must also set the <b>Use custom shared secret</b> field.

Field...	Specify...
<b>Use custom shared secret</b>	<p data-bbox="618 323 1354 390">Required only if you select the <b>Include Configuration</b> check box.</p> <ul data-bbox="618 415 1354 583" style="list-style-type: none"> <li data-bbox="618 415 1354 583">■ If you select the <b>Use custom shared secret</b> checkbox, you must provide a shared secret in the <b>Specify shared secret</b> field. Platform Manager uses the shared secret to encrypt the passwords for the configuration settings included in the snapshot template.</li> </ul> <div data-bbox="667 600 1365 800" style="background-color: #f0f0f0; padding: 5px;"> <p data-bbox="667 617 1365 789"><b>Important</b> When you apply a snapshot template that contains configuration, you must provide the same shared secret that you specified when creating the snapshot template to decrypt the passwords included in the template.</p> </div> <ul data-bbox="618 821 1354 957" style="list-style-type: none"> <li data-bbox="618 821 1354 957">■ If you clear the <b>Use custom shared secret</b> checkbox, Platform Manager uses the shared secret defined for the source installation environment or the global shared secret.</li> </ul> <p data-bbox="667 978 1354 1106">For more information about working with configurations that include passwords, see "<a href="#">Working with Configurations that Include Passwords</a>" on page 156</p>

6. Click **Save**.

Platform Manager creates the new snapshot template and transfers the files to the Command Central file system.

## Modifying a Snapshot Template

You can modify a snapshot template created with Command Central before applying the snapshot template on a target machine. For example, you can delete products, fixes, or configuration that you do not want to apply from the snapshot template. You can make changes to the snapshot template directly by editing the template files on the file system. To edit the template files on the file system, go to *Software AG\_directory*\profiles\CCE\data\templates\*template\_alias* and open the files in a text editor. You can also export the snapshot template to the client machine using the Command Central command line tool, edit the template files locally, and import the snapshot template back to the Command Central server using the Command Central command line tool.

Software AG does not recommend adding new products, fixes, files, and configurations to a created snapshot template.

**Note:** Before modifying configuration data property values, ensure that the values you specify are valid.

## Applying a Snapshot Template

**Important:** Before applying a snapshot template that contains products, you must install the Platform Manager plug-ins for the products in the installation in which you apply the template.

You can select which elements from a template of a managed installation, available in the Command Central file system, to apply to a target machine.

When you want to apply product instances included in a snapshot template, you must use the Command Central command line interface. For more information about applying templates with product instances, see [cc exec templates snapshot apply](#)

**Note:** For more information about how Command Central performs some operations and functions when you use the latest version of Command Central to create or manage a landscape with a lower version, see "[Applying Snapshot Templates](#)" on page 144.

### To apply a snapshot template to a new node

1. In the Environments pane, select the environment that contains the installation to which you want to apply a snapshot template.
2. Select the **Installations** tab.
3. On the **Installations** tab, click the installation to which you want to apply a snapshot template.
4. Click  and select **Apply template**.
5. In the Apply Template wizard, from the **Alias** column on the Template page, select the name of the snapshot template you want to apply.

You can also view the description, details, and contents of the selected snapshot template.

6. On the Products page, specify:

Field	Description
<b>Install products from template</b>	Whether to install the products included in the snapshot template. Select the check box to install products (default).
<b>Product repository</b>	The product repository from which to install the products. The list includes only product repositories configured in Command Central. For information about how to add a product

Field	Description
	repository in Command Central, see " <a href="#">Repository Management</a> " on page 285.

7. On the Fixes page, specify:

Field	Description
<b>Install products from template</b>	Whether to install the fixes included in the snapshot template. Select the check box to install fixes (default).
<b>Fix repository</b>	The repository from which to install the fixes. The list includes only fix repositories configured in Command Central. For information about how to add a fix repository in Command Central, see " <a href="#">Repository Management</a> " on page 285.

8. On the Configuration page, specify:

Field	Description
<b>Include configuration from template</b>	<p>Whether to apply the product configuration included in the snapshot template. Select the check box to apply the configuration (default).</p> <p>When this check box is selected, you also specify whether Command Central will merge or replace the existing configuration on the target node with the configuration from the snapshot template.</p>
<b>Shared secret</b>	<p>Specify the shared secret for the snapshot template. The shared secret you type in this field must match the shared secret you specified when creating the snapshot template.</p> <p>If you do not specify a shared secret, Platform Manager uses the shared secret specified for the environment to which the target node belongs or the global shared secret for the whole landscape.</p> <p>For more information about how Platform Manager determines which shared secret to</p>

Field	Description
	use, see <a href="#">"Working with Configurations that Include Passwords"</a> on page 156.
<b>Include files from template</b>	Whether to apply the files included in the snapshot template. Select the check box to apply files (default).

9. On the Confirm page, verify the snapshot template details and the apply actions you selected.
10. Click **Finish**.

Command Central applies the snapshot template to the new node to create a copy of the source installation, following the actions you specified.

## Using Default Snapshot Templates

You can also select and apply one of the default snapshot templates that Command Central provides. The following table lists the default snapshot templates available in the Apply Template wizard and their description:

Template alias	Description
um-all	Default snapshot template with products. Contains a full distribution of Universal Messaging Server with a default "umserver" instance and default HTTP port 9000.
tc-min	Default snapshot template with products. Contains a Terracotta suite distribution.  <b>Note:</b> The snapshot template contains the default tc-config.xml files. To configure a Terracotta Server Array, you must edit any custom tc-config.xml files manually. Command Central does not support configuring a Terracotta Server Array using a Platform Manager plug-in for Terracotta.
mws-min	Default snapshot template with products. Contains a default My webMethods Server instance with HTTP port 8585 and an embedded database.  <b>Note:</b> After installing the snapshot template, you can configure a connection to an external RDBMS before starting My webMethods Server for the first time, using the Command Central web user interface, command line interface, or a custom configuration snapshot template.

Template alias	Description
mws-bpm	<p>Default snapshot template with products. Contains a My webMethods Server instance for Business Process Management.</p> <p>If you apply the mws-bmp snapshot template after applying the mws-min snapshot template, Platform Manager will shut down. Software AG recommends that you install the mws-bpm snapshot template first.</p> <p><b>Note:</b> After installing the snapshot template, you can configure a connection to an external RDBMS before starting My webMethods Server for the first time, using the Command Central web user interface, command line interface, or a custom configuration snapshot template.</p>
is-min	<p>Default snapshot template with products. Contains a default Integration Server instance with HTTP port 5555 and an embedded database, but without a license file.</p> <p><b>Note:</b> When you start the default Integration Server instance for the first time, you must provide a valid Integration Server license. You add a product license using the Command Central web user interface, command line tool, or a custom configuration snapshot template.</p>
is-esb	<p>Default snapshot template with products. Contains an Integration Server instance for Enterprise Service Bus.</p> <p><b>Note:</b> When you start the default Integration Server instance for the first time, you must provide a valid Integration Server license. You add a product license using the Command Central web user interface, command line tool, or a custom configuration snapshot template.</p>
is-bpm	<p>Default snapshot template with products. Contains an Integration Server instance for Business Process Management.</p> <p><b>Note:</b> When you start the default Integration Server instance for the first time, you must provide a valid Integration Server and Business Rules license. You add a product license using the Command Central web user interface, command line tool, or a custom configuration snapshot template.</p> <p><b>Note:</b> After installing the snapshot template, you can configure a connection to an external RDBMS before starting Integration Server for the first time, using the Command</p>

Template alias	Description
	Central web user interface, command line interface, or a custom configuration snapshot template.
fix-all	<p>Default snapshot template with fixes. Contains all fixes for a product available from the Empower Product Support Website or from an image archive.</p> <p><b>Important</b> If the source fix repository includes fixes for Command Central or Platform Manager, Command Central and Platform Manager will shut down during the template application and may not start automatically. After installing the snapshot template, check if Command Central or Platform Manager are running and start them manually if required.</p>
broker-bin	<p>Default snapshot template with products. Contains only the Broker binary files, without a Broker Server instance.</p> <p>Creating a default Broker Server instance requires a valid Broker Server license file. You can create Broker Server using the Broker Server command line interface.</p>

### Considerations When Using Default Snapshot Templates

**Note:** For more information about the known limitations when applying a snapshot template, using the latest version of Command Central to create or manage a landscape with a lower version, see ["Applying Snapshot Templates" on page 144](#).

When using the default snapshot templates provided by Command Central, you must consider the following:

- Snapshot templates can be applied to an installation managed by Command Central at any time, for example right after bootstrapping a new installation or at a later time to install additional products.
- You can combine different default snapshot templates, based on the set of products that you want to install on the target installation. For example, you can apply all available default templates to a single installation in a development environment. At the same time, in a test environment you can apply the Integration Server, My webMethods Server, Universal Messaging, and Terracotta default snapshot templates to several distributed installations. When applying more than one default snapshot template to the same installation, Command Central schedules the application of the snapshot templates sequentially.
- Default snapshot templates applied to different installations on the same physical host are applied one at a time, but default snapshot templates applied to different

installations on different physical hosts will run in parallel. This insures that only one installer process is running on each machine.

- Default snapshot templates include default ports and if the same default snapshot template is applied to different installations on the same machine, you must change the default port in one or all target installations to enable running a product in all installations without port conflicts.
- Default snapshot templates do not include license files. You must add licenses for licensed products. For some products, such as Universal Messaging and Business Rules, you can add a license before starting the product, but for other products, for example Integration Server, you must start the product before adding a license.

For information about managing licenses using the Command Central web user interface or command line interface, see "[Managing Command Central Licenses and Product License Reports](#)" on page 361

- Default snapshot templates do not include configuration for an external RDBMS. When a product requires an external RDBMS connection, you must configure the connection using the Command Central web user interface, command line interface, or a configuration snapshot template created from an installation that contains a configured RDBMS connection for the product.

## Provisioning Environments

With Command Central, you can create new environments using composite templates. A composite template defines a set of environment properties for which you can specify values or use the default values provided in the template definition. For example, you can specify which products to include in a composite template or the number of nodes on which to apply the template. A composite template definition:

- Defines repositories, layers, snapshot templates, installations, and provisioning maps.
- References repositories, external files, such as license key, product, or fix image files.

If a composite template does not include references to external files, the composite template is defined in a single YAML template definition file. If a composite template contains references to external files, the composite template consists of a YAML template definition file, named `template.yaml`, and the external files referenced in the template definition, packaged in a zip archive.

With Command Central, you import and apply composite templates using the Command Central command line tool. For more information about how to use the composite templates commands, see "[Template Commands](#)" on page 338.

**Note:** When applying a composite template on the local installation, Command Central and Platform Manager will restart and the apply composite template job does not show on the jobs list.

## Composite Template Definition

A composite template is defined in the YAML format. For more information about the YAML format, see the YAML documentation. The YAML template definition file is always named “template.yaml” and located at the root level in the zip archive.

In the *Software AG\_directory\profiles\CCE\data\templates\composite* directory you can find:

- A reference template that contains information about how to define the structure of a composite template definition.
- Default composite templates that contain samples of composite template definitions for product environments. The template .yaml file for each default composite template also contains information about the input parameters that you can customize and examples of Command Central command line tool commands that you can execute for the template.

For more information about the default composite templates, see ["Using Default Composite Templates" on page 334](#)

Each template is defined with a template alias that is unique for the Command Central environment. The template declaration in the template.yaml file consists of the template alias and an optional description field. For example:

```
alias: template1
description: |
  This is a basic template
  with a description.
```

A composite template typically contains the data described in the following sections.

### Environment Parameters and Types

Environment properties are defined as input parameters under the `environments` section in the composite template definition. If an input parameter has a default value, you can use the default value or specify a user-defined value. If an input parameter does not have a default value, you must specify a user-defined value when applying the template. To specify a value for a parameter, use the following format:

```
paramName: paramValue
```

Parameter values can include references to other existing parameter values in the form of `${otherParamName}`.

```
environments:
  default:
    param1: defaultValue
    param2:
    param3: ${param1},${param2}
    param4:
  envType1:
    param2: value2
  envType2:
    param1: value1
    param3: value3
```

You can specify a type for each environment defined in a composite template in the `environment.type` parameter. In the `environments/environment.type` subsections, you specify values for the parameters required for the specific environment type, for example when you define a development environment, all parameters that you configure for your development environment are defined in the `environments/dev` subsection.

You can specify the input parameters that apply to all environments defined in a composite template in the `environments/default` section. Add the `environment/environment.type` sections when you want to customize the list of input parameters and their values for a particular environment type.

The values provided for the input parameters are resolved in the following order:

1. The values from the `environments/default` section if this section exists
2. The values from the `environments/environment.type` sections
3. The user-defined values
4. If an input parameter does not have a value, Command Central returns an error.

Example:

```
environments:
  default
    param1:defaultValue
    param2:
    param3: ${param1},${param2}
    param4: anotherValue
  envType1:
    param2:someValue
  envType2:
    param1: v1
    param3: v3
```

The values in this example environment definition are resolved as described in the following table:

Parameters	Resolved Parameter Values
no parameters	Error. Missing parameter values. Provide values for the parameters.
param2=foo	param1=defaultValue param2=foo param3=defaultValue,foo param4=anotherValue
environment.type=envType1	param1=defaultValue param2=someValue

Parameters	Resolved Parameter Values
	param3=defaultValue,someValue param4=anotherValue
environment.type=envType2 param2=foo param4=\${param2}	param1=v1 param2=foo param3=v3 param4=foo
environment.type=bar	Error. The template does not include an environment.type with value "bar".  Either provide a valid value for environment.type or leave empty to use the global default values.
param2=foo param5=bar	param1=defaultValue param2=foo param3=defaultValue,foo param4=anotherValue param5=bar
	<b>Note:</b> The value for param5 is not used in the template and is ignored.

## Layers

The `layers` section of the composite template contains the definition of the layers that will be installed on a set of nodes. A composite template should include at least one layer. Each layer is identified by an alias name and includes a ordered list of snapshot templates. The snapshot templates are applied on all nodes that map to the layer, following the exact order of the list. Each snapshot templates is defined in-line in the same composite template that contains the layer definition.

**Note:** If the template list is empty, Command Central does not apply any templates to the nodes that map to this layer.

You can also define a product or fix repository from which to install products or fixes for each layer. Optionally, you can add `layers/default` section in which you can specify a product or fix repository from which to install the products or fixes for all layers defined in the template. When a product or fix repository is defined for each individual layer,

Command Central installs products and fixes from the layer specific repositories and ignores the repository details provided in the `layers/default` section.

You can add an optional `description` for each layer.

In the following example, the composite template includes two layers with alias names “layer1” and “esb”. The templates under “layer1” and “esb” are applied on the nodes that each layer maps to, but both layers use the product and fix repositories specified in the `default` subsection.

**Example:**

```
layers:
  default:
    productRepo: webMethods-9.8
    fixRepo: Empower
  layer1:
    templates:
      - myTemlate1
      - myTemplate2
  esb:
    description: generic Integration Server layer
    templates:
      - is-server
      - is-config
```

### Provisioning Mapping

Each layer maps to one or more nodes (installations) through a provisioning map defined in the `provision` section. The nodes that a layer maps to are either existing nodes or nodes that will be created when applying the template. Each node is referenced by its global node alias.

In the `provision` section, you can map each layer to a list of nodes or you can include a `default` subsection in which you specify a default mapping that applies to environment types or nodes that do not have an explicit definition. The `provision` section can also include references to lists of host names.

In the following example, by default the “layer1” and “esb” layers map to the installation with node alias “node1”. In “envType1”, “layer1” maps to installations with aliases “node1” and “node2”. The “esb” layer maps to installations with aliases “node2” and “node3”. “envType2” maps layers to a list of user-defined host names.

**Example:**

```
provision:
  default:
    layer1: node1
    esb: node1
  envType1:
    layer1: [node1,node2]
    esb: [node2,node3]
  envType2:
    layer1: ${layer1.hosts}
    esb: ${is.hosts}
```

## Nodes

The `nodes` section defines the nodes to which the layers in the composite template are mapped.

The nodes included in this section are identified by their global alias. You can define a list of nodes for each `environment.type` included in a composite template. All parameters that you specify for a node are optional. When you do not specify parameters for a node:

- The node uses the parameters specified in the `nodes/default/default` subsection and the node alias is also used as a host name for that node.
- If the parameters are not specified in the `nodes/default/default` section, the node uses the system default parameters.

In the following example, the installations with aliases “node1” and “node2” are installed only when creating an environment with type “envType1”. The two nodes are installed on the local host with unique port numbers and installation directories. The “node3” installation is not defined with specific parameters and it maps to the parameters specified in the `nodes/default/default` subsection, using “host3” as a host name.

The definition for “node1” and “node2” does not specify user credentials. The two nodes use the “Administrator/mysecret” user credentials, defined in the `nodes/default/default` subsection.

The `nodes/default/default` subsection does not provide a value for the HTTPS port and all nodes in the template use the system default value for the HTTPS port.

### Example:

```
nodes:
  default:
    default:
      port: 8093
      secure: true
      credentials:
        username: Administrator
        password: mysecret
      bootstrapInfo:
        repoName: webMethods-US
        installDir: /opt/softwareag
        platform: lnxamd64
      credentials:
        userName: sagadmin
        password: secret123

envType1:
  node1:
    host: localhost
    port: 8192
    bootstrapInfo:
      installDir: /opt/softwareag/n1
      repoName: webMethods-EUR
  node2:
    host: localhost
    port: 8292
```

```
bootstrapInfo:
  installDir: /opt/softwareag/n2
  repoName: webMethods-EUR
```

## Snapshot Templates

You define snapshot templates in the `templates` section. The templates included in this section are identified by a unique template alias. In the `templates` section, you can include the following optional subsections:

- `products` - a list of IDs for the products that you want to install.
- `fixes` - a list of IDs for the fixes that you want to install.
- `files` - a map of files that Command Central will copy to the target installation.

In the following example, Command Central executes the following actions when applying the snapshot template with alias “is-my”:

1. Copies the license files for a specific operating system from the template archive to the target installation directory.
2. Installs Integration Server, Monitor, and Rules Engine, along with their dependent components.
3. Applies the “wMFix.integrationServer.Core.Fix3” fix.
4. Creates an Integration Server instance with alias “myInstance”, using the port and database connection details specified in the `integrationServer/myInstance` parameter.
5. Configures the “myInstance” Integration Server instance with the security credentials, included in the `integrationServer/myInstance/configuration` parameter.

### Example:

```
templates:
  is-my:
    files:
      "IntegrationServer/config/licenseKey.xml":
        "licenses/${os.platform}/PIE98.xml"
    fixes:
      - wMFix.integrationServer.Core.Fix3
    products:
      integrationServer:
        myInstance:
          primary.port: 5555
          diagnostic.port: 9999
          jmx.port: 9064
          db.type: ${db.type}
          db.url: ${db.url}
          db.username: ${db.username}
          db.password: ${db.password}
          license.file: IntegrationServer/config/licenseKey.xml
          configuration:
            integrationServer-${instance.name}:
              COMMON-LOCAL-USERS:
                COMMON-LOCAL-USERS-Administrator:
                  "@id": Administrator
                  Password: secret123
    brmspackage:
    Monitor:
```

For more information about defining a snapshot template as part of a composite template, see the reference template located in the *Software AG\_directory\profiles\CCE\data\templates\composite* directory.

## Repositories

You can define both master and image repositories for products and fixes in the `repositories` section. When you define an image repository, the repository definition must include a reference to the image file that contains products or fixes, based on the type of the repository.

You can reference repositories that are already defined in the template.

For more information about defining a repository as part of a composite template, see the reference template located in the *Software AG\_directory\profiles\CCE\data\templates\composite* directory.

For an example of a repository definition, see the `repos default` template, located in the *Software AG\_directory\profiles\CCE\data\templates\composite* directory.

## Using Default Composite Templates

Command Central provides default composite templates that you can use to set up basic or typical product environments. The default composite templates are located in the *Software AG\_directory\profiles\CCE\data\templates\composite* directory. The following table lists the default composite templates, as well as their description, environment types, and required parameters:

Template alias	Environment Types	Required Parameters	Description
bpms	<ul style="list-style-type: none"> <li>■ dev</li> <li>■ server</li> <li>■ cluster</li> </ul>	<ul style="list-style-type: none"> <li>■ database connection</li> <li>■ host names</li> </ul>	<p>Provisions a typical BPMS environment, configured with the following layers:</p> <ul style="list-style-type: none"> <li>■ messaging (Universal Messaging)</li> <li>■ cache (Terracotta Server Array)</li> <li>■ esb (Integration Server/JDBC)</li> <li>■ bpm (Integration Server/Process Engine/Rules Engine/Monitor)</li> <li>■ presentation (My webMethods Server/Task)</li> </ul>

Template alias	Environment Types	Required Parameters	Description
			<p>Engine/Rules Engine/Monitor</p> <p>Use this template to set up a general purpose BPMS environment. The template requires connection to an external relational database with an empty database schema.</p>
is-layer	<ul style="list-style-type: none"> <li>■ dev</li> <li>■ server</li> <li>■ layer</li> <li>■ cluster</li> </ul>	<ul style="list-style-type: none"> <li>■ database connection</li> <li>■ host names</li> <li>■ Terracotta Server Array URL</li> </ul>	<p>Provisions a standalone local or remote Integration Server, or an Integration Server cluster with multiple hosts.</p> <p>Use this template to setup an integration layer based on Integration Server.</p> <p>When provisioning an Integration Server cluster, you must provide an external relational database with an empty database schema and an external Terracotta layer.</p>
mws-layer	<ul style="list-style-type: none"> <li>■ dev</li> <li>■ server</li> <li>■ cluster</li> </ul>	<ul style="list-style-type: none"> <li>■ database connection</li> <li>■ host names</li> </ul>	<p>Provision a standalone local or remote My webMethods Server, or a My webMethods Server cluster with multiple hosts.</p> <p>With this template, you can setup a presentation layer using My webMethods Server.</p> <p>When provisioning a My webMethods Server cluster, you must provide an external relational database with an empty database schema.</p>

Template alias	Environment Types	Required Parameters	Description
reference	n/a	n/a	<p>Contains Domain-specific Language (DSL) documentation and reference details.</p> <p>Use this template to understand the syntax and structure of the composite template definition and to refer to examples.</p>
repos	default	Empower user name and password	<p>Contains the public master product and fix repositories version 9.5 or higher.</p> <p>These repositories are required when bootstrapping installations or executing template-provisioning operations.</p> <p>Apply this template first on the master Command Central installation that has Internet access to the Empower Product Support Website.</p>
spm-layer	<ul style="list-style-type: none"> <li>■ default</li> <li>■ dev</li> <li>■ server</li> <li>■ layer</li> </ul>	<ul style="list-style-type: none"> <li>■ SSH connection</li> <li>■ host names</li> </ul>	<p>Bootstraps a single local or remote Platform Manager, or a number of remote Platform Managers on UNIX hosts via an SSH connection.</p> <p>Use this template to setup a base management layer on which to install core products using template-based provisioning.</p>
tc-layer	<ul style="list-style-type: none"> <li>■ dev</li> <li>■ server</li> <li>■ cluster</li> </ul>	host names	Provisions a standalone local or remote Terracotta Server or a two-server

Template alias	Environment Types	Required Parameters	Description
			master/slave cluster on remote machines.
um-layer	<ul style="list-style-type: none"> <li>■ dev</li> <li>■ devCluster</li> <li>■ server</li> <li>■ cluster</li> </ul>	host names	<p>Provisions a standalone Universal Messaging realm instance or a two-instance cluster on local or remote machines.</p> <p>Use this template to set up a messaging layer using Universal Messaging.</p>

## Considerations When Using Default Composite Templates

When using the default composite templates provided by Command Central, you must consider the following:

- Before you can use the bpms and tc-layer default templates, you must replace the dummy terracotta-license.key file, located in the *Software AG\_directory*\profiles\CCE\data\templates\composite\templateAlias directory, with a valid terracotta license file. Terracotta Server does not start without a valid license.
- When using a single template for multiple target operating systems, you must create a custom bpms template and implement proper license file mapping, based on the target `os.platform` parameter. For instructions about how to create a custom composite template, see "[Creating Custom Composite Templates](#)" on page 337

## Creating Custom Composite Templates

**Important:** Do not modify the default templates provided with Command Central, other than replacing dummy license files with valid product licenses.

You can use the default composite templates, available in Command Central as a source template or example for creating custom composite templates. The default composite templates are located in the *Software AG\_directory*\profiles\CCE\data\templates\composite. Use the reference template and the default composite templates to understand the composite template YAML definition. Apply the default composite templates to understand how the bootstrapping and provisioning operations included in a composite template are executed.

### To create a custom composite template

1. Select one of the default composite templates as required.

2. Copy the template directory to a new directory. Name the new directory with a custom template alias.
3. In the new template directory, open the `template.yaml` file and rename the default template alias to the custom template alias.
4. Edit the `template.yaml` file of the custom composite template as required.

You can create a custom composite template directly on the Command Central machine by copying and modifying the files in the `Software AG_directory\profiles\CCE\data\templates\composite\alias` directory. You can also copy the default template files to a client machine. If you create a custom composite template on a client machine, you should delete and re-import the template each time you want to apply the custom template.

## Template Commands

### cc create templates snapshot

Platform Manager creates a new template based on an existing managed installation, archives the new template and transfers the archive file to the Command Central file system.

#### Syntax

- Command Central syntax:

- Use the following syntax:

With parameters:

```
cc create templates snapshot alias=template_alias
nodeAlias=node_alias options=[PRODUCTS,CONFIGURATION,FIXES,FILES]
[sharedsecret=text_string]
doOverwrite=true | false [command options]
```

With input file:

```
cc create templates snapshot
{--input | -i} filename{.xml | .json} [command options]
```

- DEPRECATED syntax:

```
cc create templates alias=template_alias
nodeAlias=node_alias options=[PRODUCTS,CONFIGURATION,FIXES,FILES]
doOverwrite=true | false [command options]
```

```
cc create templates
{--input | -i} filename{.xml | .json} [command options]
```

- Platform Manager syntax:

- With parameters:

```
cc create templates alias=template_alias
options=[PRODUCTS,CONFIGURATION,FIXES,FILES] [sharedsecret=text_string]
```

```
doOverwrite=true | false
  [command options]
```

- With input file:

```
cc create templates
  [--input | -i] filename { .xml | .json } [command options]
```

## Arguments and Options

Argument or Option	Description
<code>alias=template_alias</code>	Required. Specifies the name of the template to create.
<code>nodeAlias=node_alias</code>	Required. Only for Command Central. Specifies the alias name of the Platform Manager from which you want to create the template.
<code>options=[PRODUCTS, CONFIGURATION, FIXES, FILES]</code>	Required. Specifies the type of content to include in the new template. If you want to include more than one type of content in the template, you must repeat this parameter for each content type that you want to include.
<code>[sharedsecret=text_string]</code>	Optional. Specify a shared secret that serves as password for encrypting and decrypting passwords contained in a template with configuration.  If you do not specify a shared secret for a template, Platform Manager uses the shared secret specified for the environment to which the source installation belongs.
<code>doOverwrite=true   false</code>	Required. Specifies whether to overwrite if a template with this alias already exists. <ul style="list-style-type: none"> <li>■ <code>true</code> - overwrite the existing template</li> <li>■ <code>false</code> (default) - do not overwrite the existing template</li> </ul>
<code>{--input   -i} filename { .xml   .json }</code>	Required. Identifies an input file that contains the template metadata required for the template creation. For more information, see " <a href="#">"--input   -i"</a> on page 70.
<code>[command options]</code>	Optional. The command allows all options supported by the Command Line Interface. For

Argument or Option	Description
	a description of the options, see <a href="#">"Options for the Commands"</a> on page 59.

## Usage Notes

- The template XML metadata required for creating a template should follow the format below:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<templateMetadata>

  <!-- template to create / apply -->
  <alias>templateAlias</alias>

  <!-- node alias to create from / apply to -->
  <nodeAlias>nodeAlias</nodeAlias>

  <!-- overwrite template, if exists -->
  <overwriteTemplate>true</overwriteTemplate>

  <!-- capture / install products -->
  <productOption>
    <!-- registered product repository Id: See below -->
    <repositoryId>productRepositoryId</repositoryId>
    <type>PRODUCTS</type>
    <!-- install the latest version available in the above
         repository -->
    <useLatestVersion>>false</useLatestVersion>
  </productOption>

  <!-- capture / install fixes -->
  <fixesOption>
    <!-- registered fix repository id: See below -->
    <repositoryId>fixRepositoryId</repositoryId>
    <!-- install the latest version available in the above
         repository -->
    <useLatestVersion>true</useLatestVersion>
    <type>FIXES</type>
  </fixesOption>

  <!-- capture / copy files referenced from configurations -->
  <filesOption>
    <type>FILES</type>
  </filesOption>

  <!-- capture / apply configuration -->
  <configurationOption>
    <type>CONFIGURATION</type>
  </configurationOption>

</templateMetadata>
```

- When you make configuration changes externally to Command Central, for example by configuring settings in Integration Server Administrator, the changes may not be included in the created template. To ensure that external configuration changes are included in the template, run the `cc list configuration instances` command with the `refresh` parameter set to `true`.

- To monitor the status of the job scheduled to create the template, specify the job ID returned by the create templates command in the `cc list jobmanager jobs` command. The create templates command also provides a reference to the templates log that you can use to check the logs, using the `cc get diagnostics logs` command.
- A template that is created directly on Platform Manager is not visible to Command Central and cannot be applied to a different node. To apply a template created on Platform Manager, you must manually export the template from the node and import it into Command Central.

### Example When Executing on Command Central

- To create a template, based on the “templateMetadata.xml” file, on the Command Central server with host name “rubicon” and port “8090”:

```
cc create templates snapshot --server http://rubicon:8090/cce
-p mypassword -i D:\templateMetadata.xml
```

- To create a template with name “myTemplate” that contains the products and configuration installed on the Platform Manager node with alias “prod-is”:

```
cc create templates snapshot alias=myTemplate nodeAlias=prod-is
options=PRODUCTS options=CONFIGURATION
```

- To create a template with name “mySecureTemplate” that contains the products and configuration on the Platform Manager node with alias “prod-is” and specify “mysecret” as the shared secret for the template:

```
cc create templates snapshot alias=mySecureTemplate
nodeAlias=prod-is options=PRODUCTS options=CONFIGURATION
sharedsecret=mysecret
```

- To create a template with name “mySecureTemplate2” that contains the products and configuration on the Platform Manager node with alias “prod-is” and use the default shared secret for the environment:

```
cc create templates snapshot alias=mySecureTemplate2
nodeAlias=prod-is options=PRODUCTS options=CONFIGURATION
```

### Example When Executing on Platform Manager

To create a template, based on the “templateMetadata.xml” file, on the Platform Manager server with host name “rubicon” and port “8092”:

```
cc create templates --server http://rubicon:8092/spm -p mypassword
-i D:\templateMetadata.xml
```

### Related Commands

[cc delete templates snapshot](#)

[cc exec templates snapshot apply](#)

[cc export templates snapshot](#)

[cc list templates snapshot](#)

[cc exec templates snapshot import](#)

[cc get diagnostics logs](#)

[cc list jobmanager jobs](#)

## cc delete templates snapshot

Removes a template from a Command Central installation.

### Syntax

- Command Central syntax:

- Use the following syntax:

```
cc delete templates snapshot template_alias [options]
```

- DEPRECATED syntax:

```
cc delete templates template_alias [options]
```

```
options :  
[--force]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>template_alias</i>	Required. Specifies the template to delete. You can determine the template alias using the <code>cc list templates</code> command.
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Example When Executing on Command Central

To delete a template with name "sampleTemplate" in a Command Central installation:

```
cc delete templates snapshot sampleTemplate -p mypassword
```

### Related Command

[cc list templates snapshot](#)

## cc delete templates composite

Deletes a composite template with the specified alias.

### Syntax

- Command Central syntax:

```
cc delete templates composite template_alias [options]
```

```
options:  
[--force]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>template_alias</i>	Required. Specifies the alias of the template to delete. You can determine the template alias using the <code>cc get templates composite</code> command.
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Example When Executing on Command Central

To delete a template with name "testTemplate":

```
cc delete templates composite testTemplate -p mypassword
```

### Related Commands

[cc list templates composite](#)

## cc exec templates snapshot apply

Applies a template registered and available under the specified alias in a managed installation.

### Syntax

- Command Central syntax:

- Use the following syntax:

With parameters:

```
cc exec templates snapshot apply alias=template_alias
nodeAlias=node_alias
options=[PRODUCTS,CONFIGURATION,FIXES,FILES,INSTANCES]
[sharedsecret=text_string]
fixesRepo=fixes_repo productsRepo=products_repo
replaceConfiguration=true | false [command options]
```

**With input file:**

```
cc exec templates snapshot apply
{--input | i} filename{.xml | .json} [command options]
```

■ **DEPRECATED syntax:**

```
cc exec templates apply alias=template_alias
nodeAlias=node_alias options=[PRODUCTS,CONFIGURATION,FIXES,FILES]
fixesRepo=fixes_repo productsRepo=products_repo
replaceConfiguration=true | false [command options]
```

```
cc exec templates apply
{--input | i} filename{.xml | .json} [command options]
```

■ **Platform Manager syntax:**

■ **With parameters:**

```
cc exec templates apply alias=template_alias
options=[PRODUCTS,CONFIGURATION,FIXES,FILES,INSTANCES]
[sharedsecret=text_string]
fixesRepo=fixes_repo productsRepo=products_repo
replaceConfiguration=true | false [command options]
```

■ **With input file:**

```
cc exec templates apply
{--input | i} filename{.xml | .json} [command options]
```

**Arguments and Options**

Argument or Option	Description
<code>alias=<i>template_alias</i></code>	Required. Specifies the alias for the template to apply. You can determine the template alias using the <code>cc list templates</code> command.
<code>nodeAlias=<i>node_alias</i></code>	Required. Only for Command Central. Specifies the alias name of the Platform Manager on which you want to apply the template.
<code>options=[PRODUCTS, CONFIGURATION, FIXES, FILES, INSTANCES]</code>	Required. Specifies the type of content to apply from the template. If you want to apply more than one type of content from the template, you must repeat this parameter for each content type that you want to apply.

Argument or Option	Description
<code>[sharedsecret=<i>text_string</i>]</code>	<p>Optional. Specify a shared secret that serves as password for decrypting passwords contained in a template with configuration.</p> <p>If you do not specify a shared secret for a template, Platform Manager uses the shared secret specified for the environment to which the target installation belongs.</p>
<code>fixesRepo=<i>fixes_repo</i></code>	<p>Required when you include the <code>options=FIXES</code> parameter. Specifies the ID of the fix repository from which to install fixes.</p>
<code>productsRepo=<i>products_repo</i></code>	<p>Required when you include the <code>options=PRODUCTS</code> parameter. Specifies the ID of the product repository from which to install products.</p>
<code>replaceConfiguration= true   false</code>	<p>Required. Specifies whether to replace the configuration on the target node with the configuration from the template. When you set this parameter to true, any existing configurations on the target node that are not part of the applied template get deleted.</p> <ul style="list-style-type: none"> <li>■ <code>true</code> - replace configuration</li> <li>■ <code>false</code> (default) - do not replace configuration</li> </ul>
<code>{--input   -i} <i>filename</i>{<i>.xml</i>   <i>.json</i>}</code>	<p>Required. Identifies an input file that contains the template metadata required for applying the template. For more information, see "<a href="#">--input   -i</a>" on page 70.</p>
<code>[<i>command options</i>]</code>	<p>Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see "<a href="#">Options for the Commands</a>" on page 59.</p>

### Usage Notes

- The template XML metadata required for the template application should follow the format below:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<templateMetadata>
```

```

<!-- template to create / apply -->
<alias>templateAlias</alias>

<!-- node alias to create from / apply to -->
<nodeAlias>nodeAlias</nodeAlias>

<!-- overwrite template, if exists -->
<overwriteTemplate>true</overwriteTemplate>

<!-- capture / install products -->
<productOption>
  <!-- registered product repository Id: See below -->
  <repositoryId>productRepositoryId</repositoryId>
  <type>PRODUCTS</type>
  <!-- install the latest version available in the above
  repository -->
  <useLatestVersion>false</useLatestVersion>
</productOption>

<!-- capture / install fixes -->
<fixesOption>
  <!-- registered fix repository id: See below -->
  <repositoryId>fixRepositoryId</repositoryId>
  <!-- install the latest version available in the above
  repository -->
  <useLatestVersion>true</useLatestVersion>
  <type>FIXES</type>
</fixesOption>

<!-- capture / copy files referenced from configurations -->
<filesOption>
  <type>FILES</type>
</filesOption>

<!-- capture / apply configuration -->
<configurationOption>
  <type>CONFIGURATION</type>
  <replaceConfiguration>false</replaceConfiguration  >
</configurationOption>

</templateMetadata>

```

If `replaceConfiguration` is set to `true`, the `apply template` command replaces the existing configuration on the system on which the template is applied with the configuration included in the template. All configurations on the target system that are not part of the applied template will be removed.

- The result of the `apply template` command will look as follows:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <landscapeTemplateOperationResultDTO
    xmlns:ns2="http://www.w3.org/2005/Atom">
    <jobId>5</jobId>
    <ns2:link rel="job" href="http://localhost:8090/cce/jobmanager/jobs/5"/>
    <ns2:link rel="log" href="http://localhost:8090/cce/diagnostics/logs/
n1/OSGI-SPM/templates.log/full?search=CREATE1380025427852"/>
    <logName>CREATE1380025427852</logName>
    <nodeAlias>n1</nodeAlias>
  </landscapeTemplateOperationResultDTO>

```

- For more information about how Command Central applies snapshot templates when you use the latest version of Command Central to create or manage a landscape with a lower version, see ["Applying Snapshot Templates" on page 144](#).
- To monitor the status of the job scheduled to apply the template, specify the job ID returned by the apply templates command in the `cc list jobmanager jobs` command. The apply templates command also provides a reference to the template log that you can use to check the logs, using the `cc get diagnostics logs` command.
- When you apply a template that includes products or fixes, register and configure at least one product and fix repository on the target installation. For information about the register repository commands, see ["Repositories Commands" on page 289](#).

**Note:** Platform Manager templates are imported from Command Central as part of the apply template command. When you execute the apply template command on Platform Manager, make sure that the template already exists on this Platform Manager.

### Example When Executing on Command Central

- To apply a template, based on the "templateMetadata.xml" file, on the Command Central server with host name "rubicon" and port "8090":
 

```
cc exec templates snapshot apply --server http://rubicon:8090/ccce
-p mypassword -i D:\templateMetadata.xml
```
- To apply the products and configuration from the template with name "myTemplate" on the Platform Manager node with alias "empty-spm". Command Central installs the products from the "SAG-SDC" repository and applies the configuration from the template using the default shared secret for the environment.
 

```
cc exec templates snapshot apply alias=myTemplate nodeAlias=empty-spm
options=PRODUCTS options=CONFIGURATION
productsRepo=SAG-SDC
```
- To apply the products and configuration from the template with name "mySecureTemplate" on the Platform Manager node with alias "empty-spm". Command Central installs the products from the "SAG-SDC" repository and applies the configuration from the template, using the shared secret for the template, "mysecret123".
 

```
cc exec templates snapshot apply alias=myTemplate nodeAlias=empty-spm
options=PRODUCTS options=CONFIGURATION sharedsecret=mysecret123
productsRepo=SAG-SDC
```

### Example When Executing on Platform Manager

To apply a template, based on the "templateMetadata.xml" file, on the Platform Manager server with host name "rubicon" and port "8092":

```
cc exec templates apply --server http://rubicon:8092/spm -p mypassword
-i D:\templateMetadata.xml
```

## Related Commands

[cc create templates snapshot](#)  
[cc delete templates snapshot](#)  
[cc export templates snapshot](#)  
[cc list templates snapshot](#)  
[cc exec templates snapshot import](#)  
[cc get diagnostics logs](#)  
[cc list jobmanager jobs](#)

## cc exec templates composite apply

Applies a composite template registered and available under the specified alias in a Command Central installation.

### Syntax

- Command Central syntax:

- With input parameters:

```
cc exec templates composite apply template_alias param1=v1 param2=v2
[param3=v3...paramn=vn] [options]
```

- With input properties file:

```
cc exec templates composite apply template_alias
{--input | -i} filename.properties [options]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>template_alias</i>	Required. Specifies the alias for the template to apply. You can determine the template alias using the <code>cc list templates composite</code> command.
param1=v1 param2=v2 [param3=v3... paramn=vn]	Required. The input parameters declared in the <i>template_alias</i> YAML file.
{--input   - i} <i>filename.properties</i>	Required. A properties file that contains the input parameters declared in the <i>template_alias</i> YAML file

Argument or Option	Description
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Usage Note

- You can either include the `param1=v1 param2=v2 [param3=v3...paramn=vn]` argument or use a properties file in the command syntax. You cannot use both parameters and an input properties file in the same command.
- When applying a composite template on the local installation, Command Central and Platform Manager will restart and the apply composite template job does not show on the jobs list.

### Examples When Executing on Command Central

- A composite template with alias "myTemplate" is already imported in a Command Central installation. The template provisions an environment with type "dev" and installs a Platform Manager with alias "dev2" and port "7292" in the specified installation directory:

```
cc exec templates composite apply myTemplate environment.type=dev
spm.alias=dev2 spm.port=7292 install.dir=C:\dev\nodes\dev2
```

- To apply a template with alias "myTemplate", which is already imported in a Command Central installation, using the input parameters specified in the "mydevenv.properties" file:

```
cc exec templates composite apply myTemplate -i mydevenv.properties
```

### Related Commands

[cc list templates composite](#)

[cc exec templates composite import](#)

## cc exec templates snapshot import

Registers an exported template in a Command Central installation.

### Syntax

- Command Central syntax:

- Use the following syntax:

```
cc exec templates snapshot import [--input | -i] filename.zip
[options]
```

- DEPRECATED syntax:

```
cc exec templates import [--input | -i] filename.zip [options]
```

- Platform Manager syntax:

```
cc exec templates import [--input | -i] filename.zip [options]
```

### Arguments and Options

Argument or Option	Description
<code>{--input   -i}</code> <code>filename.zip</code>	Required. Identifies an input archive file that contains an exported template. For more information, see " <a href="#">--input   -i</a> " on page 70.
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Example When Executing on Command Central

To import a template in a Command Central installation:

```
cc exec templates snapshot import --input sampleTemplate.zip -p mypassword
```

### Example When Executing on Platform Manager

To import a template in a Command Central installation:

```
cc exec templates import --input sampleTemplate.zip -p mypassword
```

### Related Commands

[cc create templates snapshot](#)

[cc delete templates snapshot](#)

[cc exec templates snapshot apply](#)

[cc export templates snapshot](#)

[cc list templates snapshot](#)

## cc exec templates composite import

Registers an existing composite template in a Command Central installation. With this command, you can import a composite template that consists of:

- a single YAML file
- a zip archive that contains several files including the YAML template definition

### Syntax

- Command Central syntax:

```
cc exec templates composite import {--input | -i} filename.{zip| yml}
[options]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
{--input   -i} filename.{zip  yml}	Required. Identifies an input file that contains a composite template. For more information, see " <a href="#">--input   -i</a> " on page 70.
[options]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Example When Executing on Command Central

To import a composite template from the "myTemplate.zip" file into a Command Central installation:

```
cc exec templates composite import --input c:\tmp\myTemplate.zip
--input-format application/zip
```

### Related Commands

[cc exec templates snapshot apply](#)

[cc list templates composite](#)

[cc delete templates composite](#)

## cc export templates snapshot

Exports the template available under the specified alias into a compressed file.

### Syntax

- Command Central syntax:

- Use the following syntax:

```
cc get templates snapshot export template_alias
{--output | -o} filename.zip [options]
```

- DEPRECATED syntax:

```
cc get templates export template_alias
{--output | -o} filename.zip [options]
```

- Platform Manager syntax:

```
cc get templates export template_alias
```

```
{--output | -o} filename.zip [options]
```

## Arguments and Options

Argument or Option	Description
<code>template_alias</code>	Specifies the alias for the template to export. You can determine the template alias using the <code>cc list templates</code> command.
<code>{--output   -o} filename.zip</code>	Required. Specifies the output zip archive file to which to export the template. For more information about the <code>{--output   -o} file</code> option, see " <a href="#">--output   -o</a> " on page 75.
<code>[options]</code>	The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Example When Executing on Command Central

To export a template, under the alias "myAlias", from the Command Central server with host name "rubicon" and port "8090":

```
cc get templates snapshot export myAlias
--server http://rubicon:8090/cce -p mypassword
--output template-output-file.zip
```

### Example When Executing on Platform Manager

To export a template, under the alias "myAlias", from the Platform Manager server with host name "rubicon" and port "8092":

```
cc get templates export myAlias --server http://rubicon:8092/spm
-p mypassword --output template-output-file.zip
```

### Related Commands

[cc create templates snapshot](#)

[cc delete templates snapshot](#)

[cc exec templates snapshot apply](#)

[cc list templates snapshot](#)

[cc exec templates snapshot import](#)

## cc list templates composite

Retrieves a list of composite templates available in a landscape.

### Syntax

- Command Central syntax:  

```
cc list templates composite [options]
```
- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>[options]</i>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Example When Executing on Command Central

To list the composite templates on the Command Central server with host name "rubicon" and port "8090":

```
cc list templates composite --server http://rubicon:8090/cce -p mypassword
```

### Related Commands

[cc exec templates composite import](#)

[cc exec templates composite apply](#)

[cc delete templates composite](#)

## cc list templates snapshot

Retrieves a list of all templates available in a landscape.

### Syntax

- Command Central syntax:
  - Use the following syntax:  

```
cc list templates snapshot [options]
```
  - DEPRECATED syntax:  

```
cc list templates [options]
```
- Platform Manager syntax:

```
cc list templates [options]
```

## Arguments and Options

Argument or Option	Description
[options]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Example When Executing on Command Central

To list the templates on the Command Central server with host name "rubicon" and port "8090":

```
cc list templates snapshot --server http://rubicon:8090/cce -p mypassword
```

### Example When Executing on Platform Manager

To list the templates on the Platform Manager server with host name "rubicon" and port "8092":

```
cc list templates --server http://rubicon:8092/spm -p mypassword
```

### Related Commands

[cc create templates snapshot](#)

[cc delete templates snapshot](#)

[cc exec templates snapshot apply](#)

[cc export templates snapshot](#)

[cc exec templates snapshot import](#)

---

# 13 The Java Service Wrapper

---

■ The Java Service Wrapper Configuration Files .....	356
■ Specifying the JDK or JRE .....	357
■ JVM Configuration .....	357
■ Configuring the JVM in Command Central .....	358
■ The Wrapper Log File .....	359
■ Fault Monitoring .....	359
■ Generating a Thread Dump .....	360

Command Central and Platform Manager run on the Software AG Common Platform, which in turn runs in a Java Virtual Machine (JVM). The Java Service Wrapper is an application developed by Tanuki Software, Ltd.. It is a utility program that launches the JVM in which Command Central and Platform Manager run.

In addition to launching the JVM, the Java Service Wrapper offers features for monitoring the JVM, logging console output, and generating thread dumps. The following sections describe how Command Central and Platform Manager use the features of the Java Service Wrapper. For an overview of the Java Service Wrapper, see the webMethods cross-product document, *Working with the webMethods Product Suite and the Java Service Wrapper*.

## The Java Service Wrapper Configuration Files

For Command Central and Platform Manager, the configuration files for the Java Service Wrapper reside in the following directory:

- Command Central  
*Software AG\_directory/profiles/CCE/configuration*
- Platform Manager  
*Software AG\_directory/profiles/SPM/configuration*

When you start Command Central and Platform Manager, property settings in the following files determine the configuration of the JVM and the behavior of the logging and monitoring features of the Java Service Wrapper.

File name	Description
wrapper.conf	Contains property settings that are installed by Command Central and Platform Manager.  <i>Do not modify the contents of this file unless asked to do so by Software AG.</i>
custom_wrapper.conf	Contains properties that modify the installed settings in wrapper.conf.  If you need to modify the property settings for the Java Service Wrapper, you make your changes in this file.

The following sections describe configuration changes that Command Central and Platform Manager support for the Java Service Wrapper. Unless directed to do so by Software AG, do not make any configuration changes to the Java Service Wrapper other than the ones described here.

---

## Specifying the JDK or JRE

---

Command Central and Platform Manager must point to a JDK or JRE. By default, Command Central and Platform Manager point to the location of the JDK installed at the same time you installed Command Central and Platform Manager. If necessary, you can specify a different location.

**Important:** If you specify a different JDK or JRE, do not remove the JDK that Software AG Installer installed with Command Central and Platform Manager. The JDK installed with Command Central and Platform Manager is required to run the Software AG Uninstaller.

---

### To specify the Java location for Command Central and Platform Manager

1. Open the `custom_wrapper.conf` file located in:
  - For Command Central:  
`Software AG_directory/profiles/CCE/configuration`
  - For Platform Manager:  
`Software AG_directory/profiles/SPM/configuration`
2. Set the `wrapper.java.command` property so that it specifies the location of the JDK or JRE installation directory. For example:  

```
wrapper.java.command=C:\SoftwareAG\jvm\jvm\bin\java
```
3. Save and close the file.
4. Restart Command Central and Platform Manager.

---

## JVM Configuration

---

When the Java Service Wrapper launches the JVM, it provides configuration settings that, among other things, specify the size of the Java heap, the size of the PermGen area, and the directories in the classpath.

### JVM Configuration Properties

The `wrapper.java` properties in the Java Service Wrapper configuration files determine the configuration of the JVM in which Command Central and Platform Manager run.

The JVM property settings that Command Central and Platform Manager install are suitable for most environments. For additional information about the JVM property settings, see the *webMethods cross-product document, Working with the webMethods Product Suite and the Java Service Wrapper*.

Property	Value
<code>wrapper.java.initmemory</code>	Initial size (in MB) of the Java heap. The default is 32.
<code>wrapper.java.maxmemory</code>	Maximum size (in MB) to which the Java heap can grow. The default is 128.
<code>wrapper.java.classpath.n</code>	Directory in the classpath.
<code>wrapper.java.additional.n</code>	Java option to be passed in on the command line.

## Configuring the JVM in Command Central

In Command Central, you can configure JVM heap size and extended memory properties for Command Central, Platform Manager, and all managed OSGI-based products that use the Java Service Wrapper.

Extended memory properties are vendor-specific additional JVM memory options. When configuring extended properties in Command Central, you must specify only JVM memory options that follow the format:

```
-XX:optionName=value format
```

where `-XX:optionName` is the name of the extended property and `value` is the property value.

**Important:** You cannot specify JVM extended memory properties that do not have values, for example `-XX+optionName` and `-XX-optionName` are not supported.

To update JVM heap size and extended memory properties in the Command Central web user interface, locate the **Memory** configuration type under the run-time component OSGI instance, click the value for the **Initial Heap Size**, and edit the configuration details. For the steps how to manage configuration types, see ["Configuring Instances" on page 242](#).

You can also update JVM heap size and extended memory properties using the REST API or the configuration data commands in the Command Central Command Line interface. For information about working with the configuration data commands, see ["Configuring Instances" on page 242](#).

## The Wrapper Log File

---

The Java Service Wrapper records console output in a log file. The log contains the output sent to the console by the wrapper itself and by the JVM in which Command Central and Platform Manager are running. The wrapper log is especially useful when you run Command Central and Platform Manager as a Windows service, because console output is normally not available to you in this mode.

The Java Service Wrapper log is located in the following file:

- Command Central

*Software AG\_directory*\profiles\CCE\logs\wrapper.log

- Platform Manager

*Software AG\_directory*\profiles\SPM\logs\wrapper.log

To view the log, simply open the log file in a text editor.

## Logging Properties

The `wrapper.console` and `wrapper.log` properties in the wrapper configuration files determine the content, format, and behavior of the wrapper log.

The logging settings that Command Central and Platform Manager install are suitable for most environments. However, you can modify the following properties if the installed settings do not suit your needs. For procedures and additional information, see the webMethods cross-product document, *Working with the webMethods Product Suite and the Java Service Wrapper*.

<u>Property</u>	<u>Value</u>
<code>wrapper.logfile.maxsize</code>	Maximum size to which the log can grow.
<code>wrapper.logfile.maxfiles</code>	Number of old logs to maintain.
<code>wrapper.syslog.loglevel</code>	Level of messages to write to the Event Log on Windows systems or the syslog on UNIX.

## Fault Monitoring

---

The Java Service Wrapper can monitor the JVM for the certain conditions and then restart the JVM or perform other actions when it detects these conditions.

The following table describes the fault-monitoring features Command Central and Platform Manager use or allow you to configure. To learn more about these features, see the webMethods cross-product document, *Working with the webMethods Product Suite and the Java Service Wrapper*.

<b>Feature</b>	<b>Enabled</b>	<b>User configurable</b>
JVM timeout	Yes	No. Do not change the installed settings unless asked to do so by Software AG.
Deadlock detection	Yes	No. Do not change the installed settings unless asked to do so by Software AG.
Console filtering	No	No. Do not enable this feature.

## **Generating a Thread Dump**

---

The Java Service Wrapper provides a utility for generating a thread dump of the JVM when Command Central and Platform Manager are running as Windows services. A thread dump can help you locate thread contention issues that can cause thread blocks or deadlocks.

For information about generating a thread dump using the Java Service Wrapper, see the webMethods cross-product document, *Working with the webMethods Product Suite and the Java Service Wrapper*.

# 14 Managing Command Central Licenses and Product License Reports

---

■ Command Central License Overview .....	362
■ Viewing the Command Central License Information .....	362
■ Changing a Command Central License Key .....	362
■ Renewing a Command Central License Key .....	363
■ Managing Product License Reports .....	363
■ Managing License Manifests .....	365
■ License Reports Commands .....	367

## Command Central License Overview

---

When you purchase Command Central, your organization is granted a license to use it with certain features and functionality, and with a specified number of nodes to be added for administration and configuration. The license expires after a time period specified by your particular purchase agreement.

When you install Command Central, the setup program copies the license file to the *Software AG\_directory\profiles\CCE\configuration* directory with the name *cce-license.xml*.

## Viewing the Command Central License Information

---

Use the following procedure to view the license details for Command Central.

---

### To view licensing information

1. In **Command Central**, click the **Instances** tab.
2. Click **CCE**.
3. In the left pane, click **Command Central Server**.
4. Click the **Configuration** tab.
5. From the drop-down list, select **Licenses**. Command Central displays the available licenses, their status, and expiration date.
6. In the License Type column, click **Command Central**. The license location and details appear. The **PriceQuantity** field displays the number of CCE nodes allowed.
  - If the number of nodes connected is more than that specified, Command Central will shut down in 30 minutes.
  - If the number of nodes connected is equal to the one that is specified, Command Central will not allow you to add another node.
  - If the license has expired, Command Central will shut down in 30 minutes.

## Changing a Command Central License Key

---

Use the following procedure to change your license key when your license expires or if you change your license to include different features.

---

### To change the license key

1. In Command Central, click the **Instances** tab.
2. Click **CCE**.

3. In the left pane, click **Command Central Server**.
4. Click the **Configuration** tab.
5. In the **Configuration** drop-down list, select **Licenses**. Command Central displays the available licenses, their status, and expiration date.
6. In the License Type column, click **Command Central**. The license key location and details appear.
7. In the **Command Central License** page, click **Edit**.
8. Click **Browse** in the **License Upload File** section, and then navigate to the new license file.

**Note:** You will not be able to change the server license location. The new license file that you select is uploaded to the license location as shown in the **Server License Location** section.

9. Click **Save**.

## Renewing a Command Central License Key

---

If you need to obtain a new Command Central license key or renew your Command Central license, contact your Software AG sales representative.

## Managing Product License Reports

---

You can use Command Central to find out how many instances of webMethods products you have installed and monitor your license inventory. Command Central monitors a whole landscape and generates the following types of license reports of the current state of the landscape:

- Report by license key

Contains information about all installed product instances and groups them based on license key. The report also counts the server cores and compares their total with the number of license keys. The summary section of the report shows whether the customer landscape matches the available licenses. You can generate the report in XML, PDF, or JSON format, using either the Command Central web user interface or the command line tool.

- Report by license manifest

Contains information about the status of all installed product instances based on a license manifest file that contains all active licenses for a customer. The report provides detailed information per installed product, including the type of product installation, the number of installed and running product instances, the server processor types, and the type of license for the product. The report also counts the overall number of server cores and compares their total with the number of licensed

products in the license manifest. The summary section of the report shows whether the customer landscape matches the available licenses, and the number of products that match or do not match the licenses. You can generate the report in XML, PDF, or JSON format, using either the Command Central web user interface or the command line tool.

## Creating a License Report

Use the following procedure to create a license report of a landscape.

**Important:** Before creating a report by license manifest, you must add a license manifest file using the Command Central web user interface. For more information about adding a license manifest, see ["Adding License Manifests" on page 365](#).

### To create a license report

1. In Command Central, go to **Views > Licensing > Reports**.
2. Click  and select the type of report you want to create:
  - **By License Key** - Creates a license report based on a license key file.
  - **By License Manifest** - Creates a license report based on a license manifest file.
3. Click **OK**.

Command Central creates the license report and lists the report in the Licensing Reports table.

## Viewing Details About License Reports

You view the details about a license report on the Licensing Reports page in Command Central. Each license report is identified by a report ID that Command Central assigns automatically when creating the license report. You can use the report ID to search for a specific report. The Report Type column shows whether the report is generated by license key or by license manifest. The Items and Servers columns provide information about the number of licenses and servers available in a landscape. The Status column presents the status of a landscape. Valid values are:

- **Match** The landscape matches the available licenses.
- **No Match** The landscape does not match the available licenses due to any of the following reasons:
  - For the report by license key, the number of server cores exceeds the available licenses.
  - For the report by license manifest, either the number of server cores exceeds the available licenses, or the server processor type and installation type do not match the licenses.

You can also view information about when and who created the license report.

## Sorting License Reports

You can sort the items in each column of the Licensing Reports table by pointing your mouse to the column label and selecting the sorting option you require from the list.

## Downloading License Reports

You must download a license report and save it on the file system to be able to view its detailed contents.

---

### To download a license report

1. In Command Central, go to **Views > Licensing > Reports**.
2. From the Licensing Reports table, select the license report you want to download.
3. In the Download column, click , and select the format in which you want to download the report. Valid formats are: PDF, XML, and JSON.

Command Central downloads the license report file to the default download directory of the client system or a location on the client file system that you specify.

## Deleting License Reports

Use the following procedure to delete a license report.

---

### To delete a license report

1. In Command Central, go to **Views > Licensing > Reports**.
2. From the Licensing Reports table, select the license report you want to delete.
3. Click .

Command Central deletes the license report. The report is no longer available in the Licensing Reports table.

## Managing License Manifests

---

### Adding License Manifests

---

#### To add a license manifest

1. In Command Central, go to **Views > Licensing > Manifests**.
2. Click  and specify the following information:

---

Field	Description
Alias	An alias for the license manifest file.
License Manifest	The path to the license manifest file. Click <b>Browse</b> to select a license manifest file.

---

3. Click **Add**.

## Viewing Details About License Manifests

You view the details about a license manifest on the Licensing Manifests page in Command Central. Each manifest file is identified by an alias that you specify when adding the manifest file. You can use the manifest alias to search for a specific manifest.

The Location id and Location name columns provide information about the location of your organization for which the license manifest has been issued. A separate license manifest is issued for each location of your organization.

In the Status column you view the active license manifests. When a new manifest is added for an existing Location id, the old manifest for the Location Id obtains an Outdated status.

## Downloading License Manifests

You must download a license manifest file and save it on the file system to be able to view its contents.

---

### To download a license manifest

1. In Command Central, go to **Views > Licensing > Manifests**.
2. From the Licensing Manifests table, select the license manifest you want to download.
3. In the Download column, click , and select the format in which you want to download the manifest file. Valid formats are: XML and JSON.

Command Central downloads the license manifest file to the default download directory of the client system or a location on the client file system that you specify.

## Deleting License Manifests

---

### To delete a license manifest

1. In Command Central, go to **Views > Licensing > Manifests**.
2. From the Licensing Manifests table, select the license manifest you want to delete.

3. Click .

## License Reports Commands

---

### cc create license-tools reports snapshot

Creates a license inventory report by license key based on the currently registered nodes in a Command Central instance.

#### Syntax

- Command Central syntax:  

```
cc create license-tools reports snapshot [options]
```
- Not supported on Platform Manager.

#### Arguments and Options

Argument or Option	Description
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

#### Usage Notes

To check the status of the job scheduled to create the license report, use the following command:

```
cc get jobmanager jobs [jobid] --expected-values DONE --wait [seconds]
```

#### Related Commands

[cc list jobmanager jobs](#)

### cc delete license-tools reports snapshot

Deletes all generated license reports by license key from Command Central.

#### Syntax

- Command Central syntax:  

```
cc delete license-tools reports snapshot [options]
```

*options:*  

```
[--force]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>[options]</i>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Related Commands

[cc delete license-tools reports snapshot reportid](#)

[cc list license-tools reports snapshot](#)

[cc get license-tools reports snapshot reportid](#)

## cc delete license-tools reports snapshot reportid

Deletes an existing license report by license key with the specified unique report identifier.

### Syntax

- Command Central syntax:

```
cc delete license-tools reports snapshot reportid [options]
```

```
options :  
[--force]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>reportid</i>	Required. The ID of the license report to delete.
<i>[options]</i>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Example When Executing on Command Central

To delete a license report with report ID `c4eff6`:

```
cc delete license-tools reports snapshot c4eff6
```

### Related Commands

[cc delete license-tools reports snapshot](#)

[cc list license-tools reports snapshot](#)

[cc get license-tools reports snapshot reportid](#)

## cc list license-tools reports snapshot

Lists all license reports by license key available on the Command Central server.

### Syntax

- Command Central syntax:

```
cc list license-tools reports snapshot [options]
```
- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

- Command Central and Platform Manager support the following formats for the output report data:
  - comma-separated values
  - tab-separated values
  - XML
  - JSON
- The output data for each report in the list includes the report ID, the name of the user who created the report, the date the report was created, and the inventory status of the landscape.
- When an existing report fails checksum verification, the report is not included in the output list.

**Related Commands**[cc create license-tools reports snapshot](#)[cc delete license-tools reports snapshot](#)[cc delete license-tools reports snapshot reportid](#)[cc get license-tools reports snapshot reportid](#)**cc get license-tools reports snapshot reportid**

Obtains information about a license report by license key with the specified unique report identifier.

**Syntax**

- Command Central syntax:

```
cc get license-tools reports snapshot reportid [options]
```

- Not supported on Platform Manager.

**Arguments and Options**

Argument or Option	Description
<i>reportid</i>	Required. The ID of the report for which to obtain information.
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

**Examples When Executing on Command Central**

To obtain information about a report with report ID `c4eff6`:

```
cc get license-tools reports snapshot c4eff6
```

**Related Commands**[cc create license-tools reports snapshot](#)[cc delete license-tools reports snapshot](#)[cc delete license-tools reports snapshot reportid](#)[cc list license-tools reports snapshot](#)

## cc get license-tools reports snapshot output PDF

Generates a PDF file for an existing license report by license key.

### Syntax

- Command Central syntax:

```
cc get license-tools reports snapshot reportid --output-format pdf
--output filename.pdf
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>reportid</i>	Required. The ID of the license report for which to generate a PDF file.
--output-format pdf	Required. Specifies that the output is generated in PDF format.
--output- <i>filename.pdf</i>	Required. Specifies a name for the output PDF file. For more information, see " <a href="#">--output   -o</a> " on page 75.

### Usage Notes

Platform Manager sends a request to the Command Central server, with the Accept HTTP header value set to `application/pdf` to indicate that the license report output data should be generated in PDF format.

### Example When Executing on Command Central

To generate a PDF file, named `report.pdf`, for a license report with ID `c4eff6` and save the generated report file in the current directory:

```
cc get license-tools reports snapshot c4eff6 --output-format pdf
--output report.pdf
```

### Related Commands

[cc create license-tools reports snapshot](#)

[cc get license-tools reports snapshot output XML](#)

## cc get license-tools reports snapshot output XML

Generates an existing license report by license key in XML format.

### Syntax

- Command Central syntax:

```
cc get license-tools reports snapshot reportid --output-format xml
--output filename.xml
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>reportid</i>	Required. The ID of the license report for which to generate an XML file.
--output-format xml	Required. Specifies that the output is generated in XML format.
--output- <i>filename.xml</i>	Required. Specifies a name for the output XML file. For more information, see " <a href="#">--output   -o</a> " on <a href="#">page 75</a> .

### Example When Executing on Command Central

To generate an XML file, named `report.xml`, for a license report with ID `c4eff6` and save the generated report file in the current directory:

```
cc get license-tools reports snapshot c4eff6 --output-format xml
--output report.xml
```

### Related Commands

[cc create license-tools reports snapshot](#)

[cc get license-tools reports snapshot output PDF](#)

## cc create license-tools reports installation

Creates a license report of the current landscape based on a license manifest file.

### Syntax

- Command Central syntax:

```
cc create license-tools reports installation [options]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

To check the status of the job scheduled to create the license report, use the following command:

```
cc get jobmanager jobs [jobid] --expected-values DONE --wait [seconds]
```

### Related Commands

[cc delete license-tools reports installation reportid](#)

[cc get license-tools reports installation output PDF](#)

[cc get license-tools reports installation output XML](#)

[cc list license-tools reports installation](#)

## cc delete license-tools reports installation reportid

Deletes an existing license report by license manifest with the specified unique report identifier.

### Syntax

- Command Central syntax:

```
cc delete license-tools reports installation reportid [options]
```

```
options :  
[--force]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<code><i>reportid</i></code>	Required. The ID of the license report to delete.

Argument or Option	Description
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Example When Executing on Command Central

To delete a report by license manifest with report ID `c4eff7`:

```
cc delete license-tools reports installation c4eff7
```

### Related Commands

[cc create license-tools reports installation](#)

[cc delete license-tools reports installation](#)

[cc list license-tools reports installation](#)

## cc delete license-tools reports installation

Deletes all generated reports by license manifest from the Command Central server.

### Syntax

- Command Central syntax:

```
cc delete license-tools reports installation [options]
```

```
options :  
[--force]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Related Commands

[cc delete license-tools reports installation reportid](#)

[cc list license-tools reports installation](#)

## cc get license-tools reports installation output PDF

Generates a PDF file for an existing license report by license manifest.

### Syntax

- Command Central syntax:

```
cc get license-tools reports installation reportid --output-format pdf
--output filename.pdf
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>reportid</i>	Required. The ID of the license report for which to generate a PDF file.
--output-format pdf	Required. Specifies that the output is generated in PDF format.
--output <i>filename.pdf</i>	Required. Specifies a name for the output PDF file. For more information, see " <a href="#">--output   -o</a> " on page 75.

### Usage Notes

Platform Manager sends a request to the Command Central server, with the Accept HTTP header value set to `application/pdf` to indicate that the license report output data should be generated in PDF format.

### Example When Executing on Command Central

To generate a PDF file, named `report.pdf`, for a license report with ID `c4eff7` and save the generated report file in the current directory:

```
cc get license-tools reports installation c4eff7 --output-format pdf
--output report.pdf
```

### Related Commands

[cc get license-tools reports installation output XML](#)

[cc list license-tools reports installation](#)

## cc get license-tools reports installation output XML

Generates an existing license report by license manifest in XML format.

### Syntax

- Command Central syntax:

```
cc get license-tools reports installation reportid --output-format xml
--output filename.xml
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>reportid</i>	Required. The ID of the license report for which to generate an XML file.
--output-format xml	Required. Specifies that the output is generated in XML format.
--output- <i>filename.xml</i>	Required. Specifies a name for the output XML file. For more information, see " <a href="#">--output   -o</a> " on <a href="#">page 75</a> .

### Example When Executing on Command Central

To generate an XML file, named `report.xml`, for a license report with ID `c4eff7` and save the generated report file in the current directory:

```
cc get license-tools reports installation c4eff7 --output-format xml
--output report.xml
```

### Related Commands

[cc get license-tools reports installation output PDF](#)

[cc list license-tools reports installation](#)

## cc get license-tools reports installation disclaimer

Displays the disclaimer text that is valid for the reports by license manifest.

### Syntax

- Command Central syntax:

```
cc get license-tools reports installation disclaimer [options]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

---

### Related Commands

[cc list license-tools reports installation](#)

## cc list license-tools reports installation

Lists all reports by license manifest available on the Command Central server.

### Syntax

- Command Central syntax:  

```
cc list license-tools reports installation [options]
```
- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

---

### Usage Notes

- Command Central and Platform Manager support the following formats for the output report data:
  - comma-separated values
  - tab-separated values
  - XML
  - JSON

- The output data for each report in the list includes the report ID, the name of the user who created the report, the date the report was created, and the inventory status of the landscape.
- When an existing report fails checksum verification, the report is not included in the output list.

**Related Commands**

[cc list license-tools reports](#)

**cc list license-tools reports**

Lists all reports, both by license key and by license manifest, available on the Command Central server.

**Syntax**

- Command Central syntax:  

```
cc list license-tools reports [options]
```
- Not supported on Platform Manager.

**Arguments and Options**

Argument or Option	Description
<i>[options]</i>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

**Usage Notes**

- Command Central and Platform Manager support the following formats for the output report data:
  - comma-separated values
  - tab-separated values
  - XML
  - JSON
- The output data for each report in the list includes the report ID, the name of the user who created the report, the date the report was created, and the inventory status of the landscape.
- When an existing report fails checksum verification, the report is not included in the output list.

**Related Commands**

[cc list license-tools reports snapshot](#)

[cc list license-tools reports installation](#)

**cc list license-tools metering products**

Lists the names and codes of all Software AG products for which license reports are available.

**Syntax**

- Command Central syntax:

```
cc list license-tools metering products [options]
```

- Not supported on Platform Manager.

**Arguments and Options**

Argument or Option	Description
[options]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

---

**Related Commands**

[cc list license-tools reports](#)

**cc add license-tools manifests**

Adds a license manifest file with the specified manifest alias to the Command Central server.

**Syntax**

- Command Central syntax:

```
cc add license-tools manifests manifestAlias [--input|-i] filename.xml [options]
```

- Not supported on Platform Manager.

## Arguments and Options

Argument or Option	Description
<i>manifestAlias</i>	Required. The alias for the license manifest file to add.
{--input -i} <i>filename.xml</i>	Required. Specifies the name of the input file. For more information, see " <a href="#">--input   -i</a> " on page 70.
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

## Example When Executing on Command Central

To add license manifest file "Manifest1" with alias "Alias1":

```
cc add license-tools manifests alias1 --input manifest1.xml
```

## Related Commands

[cc delete license-tools manifests](#)

[cc get license-tools manifests output xml](#)

[cc list license-tools manifests](#)

## cc delete license-tools manifests

Deletes all available license manifest files or a license manifest with the specified manifest alias.

### Syntax

- Command Central syntax:

```
cc delete license-tools manifests [manifestAlias] [options] options :  
[--force]
```

- Not supported on Platform Manager.

## Arguments and Options

Argument or Option	Description
<i>manifestAlias</i>	Optional. The alias for the license manifest file to delete.

Argument or Option	Description
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Example When Executing on Command Central

To delete a license manifest with alias "Alias1":

```
cc delete license-tools manifests alias1
```

### Related Commands

[cc add license-tools manifests](#)

[cc get license-tools manifests output xml](#)

[cc list license-tools manifests](#)

## cc get license-tools manifests output xml

Generates an existing license manifest file in XML format.

### Syntax

- Command Central syntax:

```
cc get license-tools manifests manifestAlias --output-format xml
--output filename.xml[options]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>manifestAlias</i>	Required. The alias of the license manifest for which to generate an XML file.
--output-format xml	Required. Specifies that the output is generated in XML format.
--output- <i>filename.xml</i>	Required. Specifies a name for the output XML file. For more information, see " <a href="#">--output   -o</a> " on <a href="#">page 75</a> .
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For

Argument or Option	Description
	a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

---

### Example When Executing on Command Central

To generate an XML file, named manifest1.xml, for a license manifest with alias "Alias1":

```
cc get license-tools manifests alias1 --output-format xml
--output manifest1.xml
```

### Related Commands

[cc add license-tools manifests](#)

[cc delete license-tools manifests](#)

[cc list license-tools manifests](#)

## cc list license-tools manifests

Lists all license manifests available on the Command Central server.

### Syntax

- Command Central syntax:

```
cc list license-tools manifests [options]
```

- Not supported on Platform Manager.

### Arguments and Options

Argument or Option	Description
[options]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

---

### Related Commands

[cc add license-tools manifests](#)

[cc get license-tools manifests output xml](#)

[cc delete license-tools manifests](#)

# 15 Viewing Jobs

---

■ Diagnostics Logs Commands .....	384
■ Jobmanager Jobs Commands .....	391

You can view information about long-running jobs and the available job logs using the Command Central web user interface.

---

### To view information about jobs

1. In Command Central, click the **Jobs** button.

On the Jobs screen, you can view details about jobs, such as job ID, status, and progress.

2. To open a job log:
  - a. On the **Jobs** details screen, under **Log(s)** click **↓**.
  - b. Select and click the log file that you want to open.

---

## Diagnostics Logs Commands

### cc get diagnostics logs

Retrieves log entries from a log file. Log information includes the date, time, and description of events that occurred with a specified run-time component.

#### Syntax

- Command Central syntax:

You can optionally identify log(s) by supplying either a regular expression or search text.

- To optionally specify a regular expression:

```
cc get diagnostics logs node_alias runtime_componentid logid
{full | tail | head} [lines=number] [(regex=expression)
[options]
```

- To optionally specify search text:

```
cc get diagnostics logs node_alias runtime_componentid logid
{full | tail | head} [lines=number] [search=text] [options]
```

- Platform Manager syntax:

You can optionally identify log(s) by supplying either a regular expression or search text.

- To optionally specify a regular expression:

```
cc get diagnostics logs runtime_componentid logid
{full | tail | head} [lines=number] [(regex=expression) [options]
```

- To optionally specify search text:

```
cc get diagnostics logs runtime_componentid logid
{full | tail | head} [lines=number] | head [lines=number]}
[search=text] [options]
```

```

options:
[--debug | -d]
[--error | -r file]
[--log | -l file]
[--output | -o file]
[--password | -p password]
[--quiet | -q]
[--server | -s url]
[--username | -u user_name]

```

## Arguments and Options

Argument or Option	Description
<i>node_alias</i>	<p>Required. Only for Command Central. Specifies the alias name of the Platform Manager that manages the run-time component for which you want to retrieve information.</p> <p>You can determine the alias name of the Platform Manager node by using <a href="#">cc list landscape nodes</a></p>
<i>runtime_componentid</i>	<p>Required. Specifies the ID of the run-time component for which you want to retrieve information.</p> <p>You can determine the IDs for run-time components using <a href="#">cc list inventory components</a>.</p>
<i>logid</i>	<p>Required. Specifies the ID of the log for which you want to retrieve information.</p> <p>You can determine the IDs for logs using <a href="#">cc list diagnostics logs</a>.</p>
{full   tail   head}	<p>Required. Identifies the log entries you want to retrieve.</p> <ul style="list-style-type: none"> <li>■ Specify <code>full</code> to retrieve all log entries.</li> <li>■ Specify <code>tail</code> to retrieve the most recent log entries from the end of the log file.</li> <li>■ Specify <code>head</code> to retrieve entries from the beginning of the log file.</li> </ul>
[lines= <i>number</i> ]	<p>Optional. Use only with the <code>tail</code> or <code>head</code> parameters.</p>

Argument or Option	Description
	Specifies the number of log entries to return. If you omit <code>lines=number</code> the command returns 100 entries.
<code>[regex=expression]</code>	Optional. Narrows the retrieved log entries to those that meet the specified regular expression.
<code>[search=text]</code>	Optional. Narrows the retrieved log entries to those that contain the specified search text.
<code>[options]</code>	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

- By default, the command returns log entries or the full log in plain text format. If you specify a zip file for the output format, the command returns the full log file in a zip archive.
- Specify either `regex` or `search`. If you specify both, the command narrows the result using the regular expression you specify with `regex` and ignores the search text you specify with `search`.
- If you use `regex` to specify a regular expression or `search` to specify search text, and the regular expression or search text identify no log entries, the command returns no results.
- When you use `lines` with `regex` or `search`, the command returns the specified number of lines in the log that contain the specified regular expression or text. When you use `lines` without `regex` or `search`, the command returns the specified number of lines from the top or bottom of the log. For example:
  - `tail lines=10 search=JMX` returns up to ten log entries with the word "JMX".
  - `tail lines=10` returns the last ten log entries.
- When you use `{full | tail | head}` with large log files, include the `-o file` option to specify an output file. Writing a large number of log entries to the console may result in an out of memory errors.  
For more information about the `-o file` option, see [--output | -o](#).
- To avoid performance issues, do not specify a large number for `lines` when using with `tail`, `search` or `regex`.

## Examples When Executing on Command Central

- The run-time component with ID “OSGI-SPM” is managed by the Platform Manager registered as “is-dev”. The run-time component has a log with ID “default.log”. Use the following command to filter the log to entries that contain “JMX” as a word or part of a word, and return up to 20 matching entries. The results are written to the console.

```
cc get diagnostics logs is-dev OSGI-SPM default.log tail lines=20
regex=.*JMX.*
```

- The run-time component with ID “OSGI-SPM” is managed by the Platform Manager registered as “is-dev”. The run-time component has a log with ID “default.log”. Use the following command to filter the log to entries that contain the word “JMX”, and return up to 20 matching entries. The results are written to the console.

```
cc get diagnostics logs is-dev OSGI-SPM default.log head lines=20
search=JMX --password secret
```

Because the {--username | -u} option is not specified, the command uses the default user name. For more information, see “--username | -u” on page 80. The command specifies “secret” for the user’s password.

## Examples When Executing on Platform Manager

- The run-time component with ID “OSGI-SPM” is managed by the Platform Manager with host name “rubicon2” and port “8092”. The run-time component has a log with ID “default.log”. Use the following command to filter the log to entries that contain “JMX” as a word or part of a word, and return up to 20 matching entries. The results are written to the console.

```
cc get diagnostics logs OSGI-SPM default.log tail lines=20
regex=.*JMX.* --server http://rubicon2:8092/spm --password secret
```

- The run-time component with ID “OSGI-SPM” is managed by the Platform Manager with host name “rubicon2” and port “8092”. The run-time component has a log with ID “default.log”. Use the following command to filter the log to entries that contain the word “JMX”, and return up to 20 matching entries. The results are written to the console.

```
cc get diagnostics logs OSGI-SPM default.log head lines=20
search=JMX --server http://rubicon2:8092/spm --password secret
```

Because the {--username | -u} option is not specified, the command uses the default user name. For more information, see “--username | -u” on page 80. The command specifies “secret” for the user’s password.

## Related Commands

[cc list diagnostics logs](#)

[cc list inventory components](#)

## cc get diagnostic logs export file

Exports one or more log files for a specified run-time component in a zip archive file.

### Syntax

- Command Central syntax:

- To export log(s) for a specified run-time component:

```
cc get diagnostics logs node_alias runtime_componentid
[logid+logid...] export -o file [options]
```

- To export all available logs for a specified run-time component:

```
cc get diagnostics logs node_alias runtime_componentid
export -o file [options]
```

- To export logs for all run-time components:

```
cc get diagnostics logs node_alias export -o file [options]
```

- Not supported by Platform Manager.

```
options :
[--debug | -d]
[--error | -r] file
[--format | -f] file
[--log | -l] file
[--password | -p] password
[--quiet | -q]
[--server | -s] url
[--username | -u] user_name
```

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	Required. Specifies the alias name of the Platform Manager that manages the run-time component for which you want to retrieve information.
<i>runtime_componentid</i>	Required. Specifies the ID of the run-time component for which you want to retrieve information.  You can determine the IDs for run-time components using <a href="#">cc list inventory components</a> .
[ <i>logid+logid...</i> ]	A list of IDs for the log(s) that you want to export. Use the + sign as separator.

Argument or Option	Description
	You can determine the IDs for logs using <a href="#">cc list diagnostics logs</a> .
<code>-o file</code>	Required. Specifies the name of the output file. If the file you specify does not exist, the command creates it.  For more information about the output file command, see " <a href="#">--output   -o</a> " on page 75.
<code>[options]</code>	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Examples When Executing on Command Central

- The run-time component with ID "OSGI-SPM" is managed by the Command Central registered as "is-dev". The run-time component has logs with IDs "error.log" and "default.log". Use the following command to export the two logs to a zip archive file with name "test.zip".

```
cc get diagnostics logs is-dev OSGI-SPM error.log+default.log export
-o test.zip
```

- The run-time component with ID "OSGI-SPM" is managed by the Command Central registered as "is-dev". Use the following command to export all available logs for the "OSGI-SPM" component to a zip archive file with name "test.zip".

```
cc get diagnostics logs is-dev OSGI-SPM export -o test.zip
```

## cc list diagnostics logs

Lists the log files that a specified run-time component supports. Information for log files includes:

- Location of the log file
- Log ID for the log file
- Date the log file was last modified
- Size of the log file

This command returns information about the log files rather than the contents of the logs. To retrieve the contents of the log, use [cc get diagnostics logs](#).

### Syntax

- Command Central syntax:

```
cc list diagnostics logs node_alias runtime_componentid [logid]
[options]
```

### ■ Platform Manager syntax:

```
cc list diagnostics logs runtime_componentid [logid] [options]

  options:
  [--accept | -a] content_type
  [--debug | -d]
  [--error | -r] file
  [--format | -f] {xml | json}
  [--log | -l] file
  [--output | -o] file
  [--password | -p] password
  [--quiet | -q]
  [--server | -s] url
  [--username | -u] user_name
```

### Arguments and Options

Argument or Option	Description
<i>node_alias</i>	Required. Only for Command Central. Specifies the alias name of the Command Central that manages the run-time component for which you want to retrieve information.
<i>runtime_componentid</i>	Required. Specifies the ID of the run-time component for which you want to retrieve information.  You can determine the IDs for run-time components using <a href="#">cc list inventory components</a> .
[ <i>logid</i> ]	Optional. Specifies the ID of the log for which you want to retrieve information.
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

- If you do not specify the `{--format | -f}` option, the default output format is tab-separated values text.

### Examples When Executing on Command Central

- To list all log files for the run-time component that has the ID "OSGI-SPM" and is managed by the Command Central registered as "is-dev", using the authorization of the user with user name "Administrator" and password "manage", and have the information returned to the output file "loginfo":

```
cc list diagnostics logs is-dev OSGI-SPM --output
loginfo --username Administrator --password manage
```

- To list information for the log file with ID “default.log” from the run-time component that has ID “OSGI-SPM” and is managed by the Command Central registered as “is-dev”, using the authorization of the user with user name “Administrator” and password “manage”, and have the information returned to the console in JavaScript Object Notation format:

```
cc list diagnostics logs is-dev OSGI-SPM default.log
--format json --username Administrator --password manage
```

### Examples When Executing on Platform Manager

- To list all log files for the run-time component that has the ID “OSGI-SPM” and is managed by the Platform Manager with host name “rubicon2” and port “8092”, using the authorization of the user with user name “Administrator” and password “manage”, and have the information returned to the output file “loginfo”:

```
cc list diagnostics logs OSGI-SPM --server http://rubicon2:8092/spm
--output loginfo --username Administrator --password manage
```

- To list information for the log file with ID “123124” from the run-time component that has ID “OSGI-SPM” and is managed by the Platform Manager with host name “rubicon2” and port “8092”, using the authorization of the user with user name “Administrator” and password “manage”, and have the information returned to the console in JavaScript Object Notation format:

```
cc list diagnostics logs OSGI-SPM 123124 --server
http://rubicon2:8092/spm --format json --username Administrator
--password manage
```

### Related Commands

[cc get diagnostics logs](#)

## Jobmanager Jobs Commands

### cc list jobmanager jobs

Lists information about long-running jobs. A long-running job is an operation that requires more than a few seconds to complete, for example, the execution of a `cc exec lifecycle` command might take several seconds to complete.

#### Syntax

- Not supported by Command Central
- Platform Manager syntax:

```
cc list jobmanager jobs [jobid] [options]
```

```
options:
[--accept | -a] content_type]
[--debug | -d]
```

```
[{--error | -r} file]
[{--format | -f} {tsv args | xml | csv args | json}]
[{--log | -l} file]
[{--output | -o} file]
[{--password | -p} password]
[{--quiet | -q}]
[{--server | -s} url]
[{--username | -u} user_name]
```

## Arguments and Options

Argument or Option	Description
[ <i>jobid</i> ]	Optional. Specifies the ID of the job for which you want to retrieve information.
[ <i>options</i> ]	Optional. Refer to the command syntax for a list of the options the command supports. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

## Usage Notes

- If you omit [*jobid*] the command retrieves the information for all long-running jobs in the installation that Platform Manager manages.

## Examples When Executing on Platform Manager

- To retrieve information for all the long-running jobs in the installation that the Platform Manager server with host name "rubicon2" and port "8092" manages, using the authorization of the user with user name "Administrator" and password "manage":

```
cc list jobmanager jobs --server http://rubicon2:8092/spm
--username Administrator --password manage
```

- To retrieve information for the job with ID "5" that is running in the installation that the Platform Manager server with host name "rubicon2" and port "8092" manages, and have the output returned to the console in XML format:

```
cc get jobmanager jobs 3 --server http://rubicon2:8092/spm --format xml
```

Because the `{--username | -u}` and `{--password | -p}` options are not specified, the command uses the default user name and password. For more information, see ["--username | -u" on page 80](#) and ["--password | -p" on page 76](#).

## Related Commands

[cc get monitoring](#)

[cc exec lifecycle](#)

---

# III Understanding Product-specific Administration

---

■ Administering webMethods Broker .....	395
■ Administering CentraSite .....	413
■ Cloud Factory Services Overview .....	419
■ Configuring Integration Server .....	437
■ Configuring My webMethods Server .....	459
■ Configuring OSGi Profiles .....	479
■ Administering Task Engine .....	505
■ Administering Universal Messaging .....	511
■ Administering WmTaskClient .....	537
■ Administering Terracotta .....	543



---

# 16 Administering webMethods Broker

---

■ About webMethods Broker Administration .....	396
■ Configuring Broker Server License .....	396
■ Configuring SSL in Broker Server .....	397
■ Retrieving Configuration Details of Broker Server Base Port .....	398
■ Pausing and Resuming Message Publishing in Broker Servers .....	399
■ Using the Administration Link of Broker Server .....	400
■ Monitoring webMethods Broker KPIs .....	401
■ webMethods Broker and the Command Line Interface .....	405

---

## About webMethods Broker Administration

---

You can administer Broker Servers through Command Central. Note that because Platform Manager uses Broker Monitor to obtain information about Broker Servers, Broker Monitor must be running if you want to administer Broker Servers through Command Central.

You can use Command Central to perform the following operations on webMethods Broker.

- View the number of Broker Servers running in each environment of your IT landscape
- View the versions of Broker Servers
- View the fixes applied to Broker Servers
- Configure Broker Server license
- Configure SSL in a Broker Server
- Retrieve Broker Server base port and SSL configuration details
- Start, stop, and restart Broker Server
- Pause and resume message publishing in Broker Server
- Monitor Broker Server installations
- Monitor run-time status, KPIs, and alerts of Broker Server instances
- Use the administration link of Broker Server

**Note:** webMethods Broker does not support **Debug** and **Safe mode** lifecycle operations.

---

## Configuring Broker Server License

---

### To change Broker Server license

1. In the Environments pane, select the environment in which Broker Server is installed.
2. Click the **Instances** tab.
3. Click the Broker Server instance for which you want to change the license.
4. Click the **Configuration** tab.
5. Select **Licenses** from the drop-down

The license type, status, and expiration date for the Broker Server license appear below the drop-down

- In the **License Type** column, click the Broker Server link.

Command Central displays the license key location. You can view the license file details when you expand **License Key Details**.

- Click **Edit**.
- Click **Browse** in the **License Upload Location** field, and then navigate to the new license file.

The new license file that you select is uploaded to the license location as shown in the **Server License Location** field.

- If you want to change the licence file location, edit the new path in the **Server License Location** field. The new location of the server license file is updated in the `awbroker.cfg` configuration file that resides in Broker Server's data directory.
- Click **Save** to save the new license.

## Configuring SSL in Broker Server

### To enable or disable SSL in Broker Server

- In the Environments pane, select the environment in which you want to configure the Broker Server.
- Click the **Instances** tab.
- Click the name of the Broker Server instance for which you want to configure SSL.
- Click the **Configuration** tab.
- Select **Ports** in the drop-down list to view the port settings configured.  
Command Central displays the Broker Server port.
- Click the name of the port and click **Edit**.

**Connection Basics** displays the following non-editable fields.

Field	Description
<b>Enabled</b>	Whether the Broker Server base port is enabled.
<b>Port Number</b>	The Broker Server base port number.  To change the base port, stop the Broker Server and change the port setting using the <code>server_config</code> command line utility in webMethods Broker. For information about the

Field	Description
	<code>server_config</code> command line utility in webMethods Broker, see <i>Administering webMethods Broker</i> .

- Expand **Security Configuration** and specify the following SSL settings.

Field	Specify
<b>SSL Enabled</b>	Whether SSL port is enabled. Click <b>Yes</b> to enable the SSL port settings. If you do not want to use SSL, click <b>No</b> .
<b>Keystore Type</b>	The keystore type. Select the keystore type. <ul style="list-style-type: none"> <li>■ PEM</li> <li>■ PKCS12</li> </ul>
<b>Server Location of Keystore</b>	The directory where the keystore file is located.
<b>Password</b>	The password to open the keystore file.
<b>Truststore Type</b>	The truststore file format. Select the truststore type. <ul style="list-style-type: none"> <li>■ PEM</li> <li>■ DIR</li> </ul>
<b>Server Location of Truststore</b>	The directory where the truststore file is located.

- Click **Test** to verify the port settings.
- Click **Save** to save the port changes.

## Retrieving Configuration Details of Broker Server Base Port

Using Command Central, you can retrieve the configuration details of Broker Server's base port.

**Note:** You cannot use Command Central to configure the Broker Server base port. If you want to configure the base port assigned to a Broker Server, stop

the Broker Server and change the port setting using the `server_config` command line utility. For information about `server_config` command line utility, see *Administering webMethods Broker*.

### Retrieving a Broker Server's base port configuration details

1. In the Environments pane, select the environment in which Broker Server is installed.
2. Click the **Instances** tab.
3. Click the Broker Server instance for which you want to view the port settings.
4. Click the **Configuration** tab.
5. Select **Ports** from the drop-down list to view the following read-only Broker Server port details:

Field	Description
<b>Enabled</b>	Indicates whether the Broker Server port is enabled or disabled.
<b>Port</b>	Indicates the base port of the Broker Server.
<b>Protocol</b>	Indicates the protocol used by the Broker Server.
<b>Type</b>	Indicates the type of Broker Server port.

## Pausing and Resuming Message Publishing in Broker Servers

When the publishing load increases in a Broker Server, you can pause publishing, clear queues, and later resume the publishing.

### To pause and resume message publishing in a Broker Server

1. In the Environments pane, select the environment in which Broker Server is installed.
2. Select the **Instances** tab.
3. Click the status icon corresponding to the Broker Server and select the required lifecycle operation:
  - Click **Pause** to pause message publishing in all the Brokers belonging to the selected Broker Server. The status of the Broker Server changes to **Paused**. Use the  icon to refresh the status immediately. You can continue to perform

administrative tasks on paused Brokers. The clients of a paused Broker can access and retrieve the messages from the Broker queue.

- Click **Resume** to resume message publishing in all the paused Brokers belonging to the Broker Server.

The status of the Broker Server changes to **Online**. Use the  icon to refresh the status immediately.

## Using the Administration Link of Broker Server

When you have the administrative credentials to access the administration link of Broker Server in Command Central, you can use the Broker Server Details page in My webMethods.

By default, My webMethods Server running on `localhost:8585` is available for you when you click the **Broker Server Details** link. If you want to use My webMethods Server running on a different host machine, configure the host and port of the My webMethods Server you want to use.

## Configuring the Host and Port of My webMethods Server

The default host and port of the My webMethods Server specified for Broker Server administration is `localhost:8585`.

Use Command Central command line interface to configure the host and port of My webMethods Server.

## Pre-requisites for Viewing the Broker Server Details Page in My webMethods

You can access the Broker Server Details page in My webMethods only if the following conditions are true for the corresponding installation:

- My webMethods Server is installed.
- webMethods Broker user interface in My webMethods is installed.
- My webMethods Server is running.
- You have administrative credentials to access the Broker Server Details page in My webMethods.
- The Broker Server that you want to administer is added in My webMethods. For information about how to add a Broker Server in My webMethods, see *Administering webMethods Broker*.

## Viewing the Broker Server Details Page in My webMethods

### To view the Broker Server Details page in My webMethods

1. In the Environments pane, select the environment in which the Broker Server you want to administer is installed.
2. Select the **Environments > Instances** tab.
3. Click the name of the Broker Server you want to administer.
4. In the **Overview** tab, click **Broker ServerDetails**.

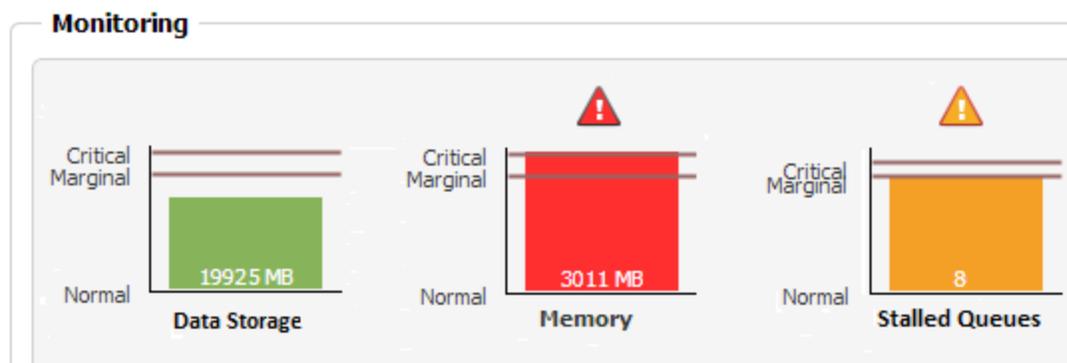
## Monitoring webMethods Broker KPIs

### Overview

The visual key performance indicators (KPIs) and alerts enable you to monitor webMethods Broker's health.

The following KPIs help you administer, troubleshoot, and resolve performance issues in webMethods Broker:

KPI	Description
<b>Data Storage or Configuration Storage</b>	Indicates the utilization of either the run-time data storage or the configuration data storage of Broker Server.
<b>Memory</b>	Indicates the utilization of Broker Server memory.
<b>Stalled queues</b>	Indicates the performance of the message queues.



## Storage Utilization KPI

Broker Server storage utilization indicator helps you to take corrective actions when either the run-time data storage or the configuration data storage of Broker Server reaches a critical value.

### Marginal, Critical, and Maximum Values for Broker Server's Storage Utilization

The marginal, critical, and maximum values of run-time data storage and configuration data storage of Broker Server depend on the maximum storage size that you have configured for the Broker Server by using the `server_config` command.

Marginal Value	Critical Value	Maximum Value
60% of the Broker Server's maximum storage size.	80% of the Broker Server's maximum storage size.	Broker Server's maximum storage size.

### Storage Utilization Display

The values of run-time data storage and configuration data storage define whether the storage indicator indicates the utilization of data storage or configuration data storage.

The storage indicator displays **Data Storage** if any of these conditions are true:

- The threshold category of both Broker Server data and configuration data storage are the same. That is, both the storage values are:
  - Less than marginal (green ■)
  - Greater than marginal but less than critical (yellow ■)
  - More than critical and less than maximum (red ■)
- Broker Server data storage has reached a higher threshold compared to configuration data storage. For example, when Broker Server data storage is at the critical threshold (yellow ■) and configuration data storage is less than marginal (green ■), then storage indicator displays the data storage value.

The storage indicator displays **Configuration Storage** if the configuration data storage has reached a higher threshold compared to the Broker Server data storage. For example, when configuration data storage is at the maximum threshold (red ■) and Broker Server data storage is more than the marginal but less than critical threshold (yellow ■), then storage indicator displays the configuration data storage value.

<u>Storage Utilization Broker Server Data</u>	<u>Storage Utilization Configuration Data</u>	<u>Storage Value Displayed</u>
Less than marginal value	Less than marginal value	Broker Server Data
Less than marginal value	More than marginal value, but less than critical value	Configuration Data
Less than marginal value	More than critical value, but less than maximum value	Configuration Data
More than marginal value, but less than critical value	Less than marginal value	Broker Server Data
More than marginal value, but less than critical value	More than marginal value, but less than critical value	Broker Server Data
More than marginal value, but less than critical value	More than critical value, but less than maximum value	Configuration Data
More than critical value, but less than maximum value	Less than marginal value	Broker Server Data
more than critical value, but less than maximum value	more than marginal value, but less than critical value	Broker Server Data
more than critical value, but less than maximum value	more than critical value, but less than maximum value	Broker Server Data

## Memory Utilization KPI

The memory utilization indicator helps you monitor Broker Server's memory.

## Marginal, Critical, and Maximum Values for Memory Utilization

The marginal, critical, and maximum values of memory utilization depend on the Broker Server's memory defined by the `max-memory-size` parameter in the Broker Server configuration file (`awbroker.cfg`).

Marginal Value	Critical Value	Maximum Value
80% of the <code>max-memory-size</code> parameter value.	95% of the <code>max-memory-size</code> parameter value.	Broker Server's memory limit defined in the <code>max-memory-size</code> parameter.

## Stalled Queues KPI

The stalled queues indicator alerts you if messages are stuck for a long time or if messages are never retrieved from queues that are connected to clients.

A queue is considered to be stalled only if all these conditions are true:

- A client is connected to the queue
- The queue contains at least one message
- It has been more than five minutes since the client retrieved a message from the queue

Marginal Value	Critical Value	Maximum Value
1 queue.	50% of the maximum value.	Defined by whichever of these values is greater: <ul style="list-style-type: none"> <li>■ 1 queue.</li> <li>■ 5% of the total number of client queues or forward queues in Brokers.</li> <li>■ Current number of stalled queues.</li> </ul> For example, if the number of stalled queues is zero, and 5% of the sum of client queues and forward queues is less than 1, then the

**Marginal Value****Critical Value****Maximum Value**

maximum value is 1 queue.

## webMethods Broker and the Command Line Interface

### Commands that webMethods Broker Supports

webMethods Broker supports the Platform Manager commands listed in the following table. The table lists where you can find general information about each command. Additionally, if there is webMethods Broker-specific information, the table lists where you can learn more about arguments and options that webMethods Broker supports or details about the actions Broker takes when you execute an `exec` command.

Commands	Additional Information
<code>cc get configuration data</code>	<p>For general information about the command, see <a href="#">"cc get configuration data" on page 253</a>.</p> <p>For webMethods Broker-specific information about using the command, see <a href="#">"Configuration Types that webMethods Broker Supports" on page 407</a>.</p>
<code>cc update configuration data</code>	<p>For general information about the command, see <a href="#">"cc update configuration data" on page 256</a>.</p>
<code>cc get configuration instances</code>	<p>For general information about the command, see <a href="#">"cc get configuration instances" on page 260</a>.</p> <p>For webMethods Broker-specific information about using this command, see <a href="#">"Configuration Types that webMethods Broker Supports" on page 407</a>.</p>
<code>cc list configuration instances</code>	<p>For general information about the command, see <a href="#">"cc list configuration instances" on page 263</a>.</p> <p>For webMethods Broker-specific information about using this command,</p>

Commands	Additional Information
	see <a href="#">"Configuration Types that webMethods Broker Supports"</a> on page 407.
cc get configuration types	<p>For general information about the command, see <a href="#">"cc get configuration types"</a> on page 265.</p> <p>For webMethods Broker-specific information about using this command, see <a href="#">"Configuration Types that webMethods Broker Supports"</a> on page 407.</p>
cc list configuration types	<p>For general information about the command, see <a href="#">"cc list configuration types"</a> on page 267.</p> <p>For webMethods Broker-specific information about using this command, see <a href="#">"Configuration Types that webMethods Broker Supports"</a> on page 407.</p>
cc get inventory components	For general information about the command, see <a href="#">"cc get inventory components"</a> on page 194.
cc list inventory components	For general information about the command, see <a href="#">"cc list inventory components"</a> on page 198.
cc exec lifecycle	<p>For general information about the command, see <a href="#">"cc exec lifecycle"</a> on page 236.</p> <p>For webMethods Broker-specific information about using this command, see <a href="#">"Lifecycle Actions for Broker Server "</a> on page 408.</p>
cc get monitoring	<p>For general information about the command, see <a href="#">"cc get monitoring"</a> on page 227.</p> <p>For webMethods Broker-specific information about using this command, see:</p> <ul style="list-style-type: none"> <li>■ <a href="#">"Monitoring Run-time Statuses for webMethods Broker "</a> on page 409</li> <li>■ <a href="#">"Monitoring Run-time States for webMethods Broker "</a> on page 410</li> </ul>

## Configuration Types that webMethods Broker Supports

The following table lists the configuration types that webMethods Broker supports.

Configuration Type	Use to configure...
BROKER-MWSADMIN	<p>The protocol, host, and port of the My webMethods Server that hosts the administration user interface for the Broker for which the command was executed. The default value is <code>http://localhost:8585</code>.</p> <p>You can use the following commands to retrieve or update the value:</p> <ul style="list-style-type: none"> <li>■ <a href="#">cc get configuration data</a></li> <li>■ <a href="#">cc update configuration data</a></li> </ul>
COMMON-ADMINUI	<p>The full URL for the Broker Server Details page in My webMethods. You cannot change the value. You can use the <a href="#">cc get configuration data</a> to retrieve its value.</p>
COMMON-LICENSE	<p>The SagLic license file. You can use the following commands to add and update webMethods Broker-specific information to the SagLic file:</p> <ul style="list-style-type: none"> <li>■ <a href="#">cc get configuration data</a></li> <li>■ <a href="#">cc update configuration data</a></li> <li>■ <a href="#">cc get configuration instances</a></li> <li>■ <a href="#">cc list configuration instances</a></li> <li>■ <a href="#">cc get configuration types</a></li> <li>■ <a href="#">cc list configuration types</a></li> </ul>
COMMON-LICLOC	<p>The location where the license file resides in file system where Broker Server is installed.</p>
COMMON-PORTS	<p>The Broker Server listener port. You can use the following commands to retrieve or update port information:</p> <ul style="list-style-type: none"> <li>■ <a href="#">cc get configuration data</a></li> <li>■ <a href="#">cc update configuration data</a></li> </ul>
	<p><b>Note:</b> You cannot use the command line interface to change the Broker Server port number. You can only change the SSL information.</p>

### Example When Executing on Command Central

To change the URL of the My webMethods Server that hosts the Broker Server administration user interface to `http://localhost:8500`, do the following:

1. Read the current configuration of BROKER-MWSADMIN of the Broker-Server-8349 instance and store the configuration details in the BROKER-MWSADMIN.txt file:

```
cc get configuration data node_alias Broker-Server-8349
BROKER-MWSADMIN -p password -o BROKER-MWSADMIN.txt
```

2. Using a text editor, edit the BROKER-MWSADMIN.txt file to change the URL to `http://localhost:8500`.

3. Update BROKER-MWSADMIN using the new settings in the BROKER-MWSADMIN.txt file:

```
cc update configuration data node_alias Broker-Server-8349
BROKER-MWSADMIN -p password -i BROKER-MWSADMIN.txt
```

4. View the My webMethods Server URL change in the BROKER-MWSADMIN configuration:

```
cc get configuration data node_alias Broker-Server-8349
BROKER-MWSADMIN -p password -o BROKER-MWSADMIN.txt
```

5. Refresh the COMMON-ADMINUI configuration:

```
cc get configuration data node_alias Broker-Server-8349
COMMON-ADMINUI refresh=true -p password
```

### Related Commands

[cc get configuration instances](#)

[cc list configuration instances](#)

[cc get configuration types](#)

[cc list configuration types](#)

## Lifecycle Actions for Broker Server

The following table lists the actions that webMethods Broker supports with the `cc exec lifecycle` command and the operation taken against Broker Server when an action is executed.

Action	Description
start	Starts Broker Server. When successful, the Broker Server run-time status is set to ONLINE.
stop	Stops Broker Server. The Broker Server run-time status is STOPPED.

Action	Description
restart	Stops, then restarts Broker Server. The Broker Server run-time status is set to ONLINE.
pause	<p>Pauses operation on Broker Server. The Broker Server run-time status is set to PAUSED.</p> <p>When a Broker Server is paused, all Brokers that belong to the Broker Server stop publishing documents. However, Broker clients can still retrieve documents from the client queues, and you can still perform administrative tasks, such as creating clients and document types on the paused Brokers.</p>
resume	Resumes a previously paused Broker Server. As a result, Brokers that belong to the Broker Server start publishing documents again. The Broker Server run-time status is returns to ONLINE.

### Related Commands

[cc exec lifecycle](#)

## Monitoring Run-time Statuses for webMethods Broker

The following table lists the run-time statuses that webMethods Broker can return in response to the `cc get monitoring runtimestatus` and `cc get monitoring state` commands, along with the meaning of each run-time status.

Run-time Status	Meaning
ONLINE	Broker Server is running.
FAILED	Broker Server is not running due to some failure, and attempts to start it again have failed.
PAUSED	All Brokers belonging to the Broker Server have paused document publishing.
STARTING	Broker Server is starting.
STOPPED	Broker Server is not running because it was shut down normally.

Run-time Status	Meaning
STOPPING	Broker Server is stopping.
UNKNOWN	The status of Broker Server cannot be determined.
UNRESPONSIVE	Broker Server does not respond to a ping operation. However, other indicators, such as the existence of the PID or LOCK file indicate that Broker Server is running.

### Related Commands

[cc get monitoring](#)

## Monitoring Run-time States for webMethods Broker

In response to the `cc get monitoring runtimestate` and `cc get monitoring state` commands, webMethods Broker provides information about the following key performance indicators (KPIs):

KPI	Description
Data storage or configuration storage usage	<p>Use this KPI to monitor the Broker Server run-time memory storage or configuration storage so that you can take corrective actions if storage use approaches a critical value.</p> <p>The marginal, critical, and maximum values for this KPI depend on the maximum storage size configured for Broker Server.</p> <ul style="list-style-type: none"> <li>■ Marginal is 60% of the maximum storage size.</li> <li>■ Critical is 80% of the maximum storage size.</li> <li>■ Maximum is the configured storage size.</li> </ul>
Memory usage	<p>Use this KPI to monitor use of Broker Server main memory so that you can take corrective actions if memory use approaches a critical value.</p> <p>The marginal, critical, and maximum values for this KPI depend the amount of main memory configured in the Broker Server configuration file (<code>awbroker.cfg</code>).</p> <ul style="list-style-type: none"> <li>■ Marginal is 80% of the maximum main memory.</li> <li>■ Critical is 95% of the maximum main memory.</li> </ul>

KPI	Description
	<ul style="list-style-type: none"> <li>■ Maximum is the configured amount of main memory.</li> </ul>
Stalled queues	<p>Use this KPI to monitor stalled queues. A queue is considered stalled only if all the following conditions are true for that queue:</p> <ul style="list-style-type: none"> <li>■ A client is connected to the queue.</li> <li>■ The queue contains at least one message.</li> <li>■ More than five minutes has elapsed since the last time the client retrieved a message from the queue.</li> </ul> <p>The KPI uses the following marginal, critical, and maximum values:</p> <ul style="list-style-type: none"> <li>■ Marginal is 1 stalled queue.</li> <li>■ Critical is 50% of the maximum number of queues.</li> <li>■ Maximum is determined by the greatest value among the following conditions:             <ul style="list-style-type: none"> <li>■ 1 queue</li> <li>■ 5% of the total number of client or forward queues in all Brokers.</li> <li>■ Current number of stalled queues.</li> </ul> </li> </ul>

For more information about the webMethods Broker KPIs, see information about monitoring webMethods Broker in the "[Monitoring webMethods Broker KPIs](#)" on page 401.

### Related Commands

[cc get monitoring](#)



# 17 Administering CentraSite

---

■ About Administering CentraSite .....	414
■ Viewing CentraSite Components .....	414
■ CentraSite and the Command Line Interface .....	415

## About Administering CentraSite

---

CentraSite installation contains two components:

- CentraSite Registry Repository (CRR)
- CentraSite Application Server Tier (CAST)

You can use Command Central to perform the following administration tasks on your CentraSite installation:

- View the CentraSite components. For more information, see [Viewing CentraSite Components](#).
- Start, stop, restart, and debug the CentraSite Registry Repository. If you start or restart CentraSite Registry Repository when it is on debug mode, the debug mode turns off, and CentraSite Registry Repository works on normal mode.

CentraSite Application Server Tier, a component of CentraSite, runs in the Software AG Runtime. You cannot start, stop, or restart the CentraSite Application Server Tier independent of the Software AG Runtime (CTP).

- Create log files for debugging. When you perform the Debug action on the CentraSite Registry Repository, CentraSite writes status and other information to the following log files in the *CentraSite\_directory*\data directory, where *CentraSite\_directory* is the installation directory of CentraSite.

Log File...	Stores...
Registry.log (CentraSite.AAB.log.1.xml CentraSite.AAB.log.0.xml)	The request logs (middle level information).
A file with type "2X0"	The data store request logs (low level information).

For more information about how to start, stop, restart, and debug a CentraSite Registry Repository, see [Starting, Stopping, Pausing, Resuming, and Debugging Instances](#).

## Viewing CentraSite Components

---

### Viewing CentraSite Registry Repository (CRR) and CentraSite Application Server Tier (CAST)

CentraSite Registry Repository is a process.

CentraSite Application Server is an engine.

### To view CRR and CAST

1. In the Environments pane, select the environment that contains the CentraSite installation you want to view.
2. Click the **Instances** tab.
3. To view CRR, click **CentraSiteRegistry Repository**. If there is more than one CRR instance, click the one that you want to work with.
4. To view CAST, expand the **CTP** node.
5. Click **CentraSiteApplication Server**.

## CentraSite and the Command Line Interface

### Commands that CentraSite Registry Repository Supports

The CentraSite Registry Repository run-time component supports the Platform Manager commands listed in the following table. The table lists where you can find general information about each command. Additionally, the table lists where you can learn more about arguments and options that the run-time component supports or details about the actions it takes when you execute an `exec` command.

Commands	For more information, see...
<code>cc get inventory components</code>	For general information about the command, see <a href="#">"cc get inventory components" on page 194</a> .
<code>cc list inventory components</code>	For general information about the command, see <a href="#">"cc list inventory components" on page 198</a> .
<code>cc exec lifecycle</code>	For general information about the command, see <a href="#">"cc exec lifecycle" on page 236</a> .  For CentraSite-specific information about using this command, see <a href="#">"Lifecycle Actions for CentraSite Registry Repository" on page 416</a> .
<code>cc get monitoring</code>	For general information about the command, see <a href="#">"cc get monitoring" on page 227</a> .

Commands	For more information, see...
	For CentraSite-specific information about using this command, see <a href="#">"Run-time Monitoring Statuses for CentraSite Registry Repository"</a> on page 417.

## Commands that CentraSite Application Server Tier Supports

The CentraSite Application Server Tier run-time component supports the Platform Manager commands listed in the following table. The table lists where you can find general information about each command.

Commands	For more information, see...
<code>cc get inventory components</code>	For general information about the command, see <a href="#">"cc get inventory components"</a> on page 194.
<code>cc list inventory components</code>	For general information about the command, see <a href="#">"cc list inventory components"</a> on page 198.

**Note:** The CentraSite Application Server Tier does not support the `cc exec lifecycle` command. The CentraSite Application Server Tier is a CentraSite component, but it runs inside Software AG Runtime. You cannot start, stop, or restart the CentraSite Application Server Tier independent of the Software AG Runtime (CTP).

## Lifecycle Actions for CentraSite Registry Repository

The following table lists the actions that the CentraSite Registry Repository (CRR) run-time component supports with the `cc exec lifecycle` command and the operation taken against the run-time component when an action is executed.

Action	Description
<code>start</code>	Starts the run-time component. When successful, the run-time status is set to ONLINE.
<code>stop</code>	Stops the run-time component. When successful, the run-time status is set to STOPPED.

Action	Description
	<b>Note:</b> The CentraSite Server waits for currently active processing to finish before stopping.
<code>restart</code>	Stops, then restarts the run-time component. The run-time status is set to ONLINE.
<code>startindebugmode</code>	Starts the run-time component in debug mode. When successful, the run-time status is set to ONLINE.  While running in Debug mode, CentraSite writes status and other information to log files in the <i>CentraSite_directory\data</i> directory.

### Related Commands

[cc exec lifecycle](#)

## Run-time Monitoring Statuses for CentraSite Registry Repository

The following table lists the run-time statuses that the CentraSite Registry Repository (CRR) run-time component can return in response to the `cc get monitoring runtimestatus` and `cc get monitoring state` commands, along with the meaning of each run-time status.

Run-time Status	Meaning
ONLINE	The run-time component is running.
STARTING	The run-time component is starting.
STOPPED	The run-time component is not running because it was shut down normally.
STOPPING	The run-time component is stopping.
UNKNOWN	The status of the run-time component cannot be determined.

### Related Commands

[cc get monitoring](#)



# 18 Cloud Factory Services Overview

---

■ Configuring Cloud Factory Services Accounts .....	420
■ Cloud Factory Services Commands .....	424

With Software AG Cloud Factory Services, you can access services provided by different cloud service providers using a common interface, which helps reduce the complexity of vendor-specific library dependencies and cloud operations. Cloud Factory Services is an abstraction layer over the APIs of cloud service providers.

Using Command Central, you can manage Cloud Factory Services accounts for the following vendors:

- Amazon Elastic Compute Cloud (Amazon EC2)
- Rackspace
- VMware vSphere

For information about configuring Cloud Factory Services accounts, see "[Configuring Cloud Factory Services Accounts](#)" on page 420.

## Configuring Cloud Factory Services Accounts

---

### Adding an Amazon Elastic Compute Cloud Account

You add an Amazon Elastic Compute Cloud (Amazon EC2) account that you use with the Cloud Factory Services API using the Command Central web user interface.

**Important:** You must obtain an Amazon EC2 secret key and access key from the Amazon Web Services (AWS) management console before adding an Amazon EC2 account.

**Note:** All Cloud Factory Services instances should be created from images (templates) with predefined user names and passwords because Cloud Factory Services does not support key generation.

---

#### To add an Amazon EC2 account

1. Select the Command Central environment from the Environment pane, then click the instance from the **Instances** tab.
2. Select the Cloud Factory component.
3. Click the **Configuration** tab.
4. Select **Accounts** in the drop-down list.

Command Central displays the Cloud Factory Services account configuration page.

5. Click  to add an account.

Command Central displays the Select Configuration dialog box.

6. Select **EC2** from the drop-down list and click **OK**.
7. Complete the following fields:

<b>Field</b>	<b>Specify</b>
<b>Account Name</b>	The name for the Amazon EC2 account. Do not use spaces in the account name.
<b>Vendor</b>	The name of the account vendor. Command Central generates the name automatically based on the configuration type you selected.
<b>Region</b>	The geographic region for which the account is created.
<b>Availability Zone</b>	Optional. The Amazon EC2 location for which the account is created.
<b>Secret Key</b>	The secret key provided by Amazon EC2.
<b>Access Key</b>	The access key provided by Amazon EC2.
<b>Proxy Host</b>	Optional. The proxy host name if the operating system uses a proxy.
<b>Proxy Port</b>	Optional. The number of the proxy port if the operating system uses a proxy.

8. Click **Test** to test the configuration for the account.
9. Click **Save**.

## Adding a Rackspace Account

You add a Rackspace account that you use with the Cloud Factory Services API using the Command Central web user interface.

**Note:** All Cloud Factory Services instances should be created from images (templates) with predefined user names and passwords because Cloud Factory Services does not support key generation.

### To add a Rackspace account

1. Select the Command Central environment from the Environment pane, then click the instance from the **Instances** tab.
2. Select the Cloud Factory component.
3. Click the **Configuration** tab.

4. Select **Accounts** in the drop-down list.

Command Central displays the Cloud Factory Services account configuration page.

5. Click  to add an account.

Command Central displays the Select Configuration dialog box.

6. Select **Rackspace** from the drop-down list and click **OK**.

7. Complete the following fields:

Field	Specify
<b>Account Name</b>	The name for the Rackspace account. Do not use spaces in the account name.
<b>Vendor</b>	The name of the account vendor. Command Central generates the name automatically based on the configuration type you selected.
<b>Endpoint</b>	Depending on the region in which the servers are located, values are: <ul style="list-style-type: none"> <li><input type="checkbox"/> uk</li> <li><input type="checkbox"/> us</li> </ul>
<b>Region</b>	The geographic region in which the servers are located.
<b>Username</b>	The user name for logging on the Rackspace account.
<b>Password</b>	The password for logging on the Rackspace account.
<b>Proxy Host</b>	Optional. The proxy host name if the operating system uses a proxy.
<b>Proxy Port</b>	Optional. The number of the proxy port if the operating system uses a proxy.

8. Click **Test** to test the configuration for the account.

9. Click **Save**.

## Adding a VMware vSphere Account

You add a VMware vSphere account that you use with the Cloud Factory Services API using the Command Central web user interface.

**Important:** You must install a vSphere SDK before adding a VMware vSphere account.

### To add a VMware vSphere account

1. Select the Command Central environment from the Environment pane, then click the instance from the **Instances** tab.
2. Select the Cloud Factory component.
3. Click the **Configuration** tab.
4. Select **Accounts** in the drop-down list.  
Command Central displays the Cloud Factory Services account configuration page.
5. Click  to add an account.  
Command Central displays the Select Configuration dialog box.
6. Select **VSphere** from the drop-down list and click **OK**.
7. Complete the following fields:

Field	Specify
<b>Account Name</b>	The name for the VMware vSphere account. Do not use spaces in the account name.
<b>Vendor</b>	The name of the account vendor. Command Central generates the name automatically based on the configuration type you selected.
<b>Endpoint</b>	The endpoint URL for the VMware vSphere SDK.
<b>Region</b>	The name of the data center accessed by this account. If you do not specify a value, the account connects to all machines to which you are granted access.
<b>Username</b>	The user name for logging on the VMware vSphere account.
<b>Password</b>	The password for logging on the VMware vSphere account.

8. Click **Test** to test the configuration for the account.
9. Click **Save**.

## Editing Accounts

You can update the details for an account registered with the Cloud Factory Services API using the Command Central user interface.

---

### To edit an account registered with the Cloud Factory Services API

1. Select the Command Central environment from the Environment pane, then click the instance from the **Instances** tab.
2. Select the Cloud Factory component.
3. Click the **Configuration** tab.
4. Select **Accounts** in the drop-down list.  
Command Central displays the Cloud Factory Services account configuration page.
5. Click the name of the account that you want to update.
6. On the Configuration details page for the account, click **Edit**.
7. Update the required fields.
8. Click **Save**.

## Deleting Accounts

You can use the Command Central web user interface to delete a registered Cloud Factory Services account.

---

### To delete a Cloud Factory Services account

1. Select the Command Central environment from the Environment pane, then click the instance from the **Instances** tab.
2. Select the Cloud Factory component.
3. Click the **Configuration** tab.
4. Select **Accounts** from the drop-down list.  
Command Central displays the Cloud Factory Services account configuration page.
5. Select an account and click  to delete the account.

---

## Cloud Factory Services Commands

### cc create cfs instances

Creates an instance for a Cloud Factory Services account.

#### Syntax

- Command Central syntax:

```
cc create cfs instances accountName
imageId
[type={instanceType | instanceFlavor}]
```

```
[firewall={securityGroup | serverName }]  
[options]
```

- Not supported by Platform Manager.

## Arguments and Options

Argument or Option	Description
<i>accountName</i>	Required. Specifies the name of a Cloud Factory Services account.
<i>imageId</i>	Required. Specifies the image ID from which to create a new instance.
[type= {instanceType   instanceFlavor }]	Depending on the account: <ul style="list-style-type: none"> <li>■ For Amazon EC2 accounts, specifies the instance type. For information about the supported instance types, see the Amazon EC2 documentation.</li> <li>■ For Rackspace accounts, specifies the instance flavor. For information about the supported instance flavors, see the Rackspace documentation.</li> <li>■ Not supported for VMware vSphere accounts.</li> </ul>
[firewall= {securityGroup   serverName }]	Depending on the account: <ul style="list-style-type: none"> <li>■ For Amazon EC2 accounts, specifies the name of an Amazon EC2 security group. For information about security groups, see the Amazon EC2 documentation.</li> <li>■ For Rackspace accounts, specifies the name of a Rackspace server. For information about specifying Rackspace server names, see the Rackspace documentation.</li> <li>■ Not supported for VMware vSphere accounts.</li> </ul>
[options]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

## Usage Notes

Instead of specifying the `{--server|-s}` option in the command, you can set the `CC_SERVER` environment variable to the endpoint URL of the Command Central server that hosts Cloud Factory Services as follows:

On Windows: `set CC_SERVER=http://host:port`

On UNIX: `export CC_SERVER="http://host:port/"`

where *host* is the name and *port* is the port of the Command Central server that hosts Cloud Factory Services.

**Note:** The endpoint URL for the Command Central server on which Cloud Factory Services is installed contains *only* host name and port.

For information about `{--server|-s}`, see "[--server | -s](#)" on page 77.

## Example When Executing on Command Central

To create an instance for an Amazon EC2 account with name "ec2bvt" from an image with ID "ami-1f918276", and instance type "m1.small", on a Command Central server with host name "rubicon" and port "8490" that is behind a firewall with name "cc":

```
cc create cfs instances ec2bvt ami-1f918276
type=m1.small firewall=cc --server http://rubicon:8490
```

To check if the new instance is created and to obtain the run-time status for the created instance, use the `cc get cfs instances accountName instanceId {--expected-values|-e} expectedStatus` command.

To obtain the run-time status of an instance with ID "i-7b40ab2b" for an Amazon EC2 account with name "ec2bvt" on a Command Central server with host name "rubicon" and port "8490":

```
cc get cfs instances ec2bvt i-7b40ab2b
--server http://rubicon:8490 -e running
```

For information about `--expected-values|-e`, see "[--expected-values | -e](#)" on page 64.

## Related Commands

[cc delete cfs instances](#)

[cc exec cfs instances](#)

[cc get cfs instances](#)

[cc list cfs accounts](#)

[cc list cfs images](#)

[cc list cfs instances](#)

## cc delete cfs instances

Deletes an instance for a Cloud Factory Services account.

### Syntax

- Command Central syntax:

```
cc delete cfs instances accountName instanceId [options]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>accountName</i>	Required. Specifies the name of a Cloud Factory Services account.
<i>instanceId</i>	Required. Specifies the ID of the VM instance you want to delete.
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

Instead of specifying the `{--server|-s}` option in the command, you can set the `CC_SERVER` environment variable to the endpoint URL of the Command Central server that hosts Cloud Factory Services as follows:

On Windows: `set CC_SERVER=http://host:port`

On UNIX: `export CC_SERVER="http://host:port/"`

where *host* is the name and *port* is the port of the Command Central server that hosts Cloud Factory Services.

**Note:** The endpoint URL for the Command Central server on which Cloud Factory Services is installed contains *only* host name and port.

For information about `{--server|-s}`, see ["--server | -s" on page 77](#).

### Example When Executing on Command Central

To delete a VM instance with ID "i-7b40ab2b" for an Amazon EC2 account with name "ec2" on a Command Central server with host name "rubicon" and port "8490":

```
cc delete cfs instances ec2 i-7b40ab2b --server http://rubicon:8490
```

To check if the instance is deleted, use the `cc`

```
get cfs instances accountName instanceId {--expected-values|-e} expectedStatus command.
```

For example, to obtain the status of an instance with ID “i-7b40ab2b” for an Amazon EC2 account with name “ec2” on a Command Central server with host name “rubicon” and port “8490”:

```
cc get cfs instances ec2 i-7b40ab2b
--server http://rubicon:8490 -e terminated
```

For information about `--expected-values|-e`, see [“--expected-values | -e” on page 64](#).

### Related Commands

[cc create cfs instances](#)

[cc exec cfs instances](#)

[cc get cfs instances](#)

[cc list cfs accounts](#)

[cc list cfs images](#)

[cc list cfs instances](#)

## cc exec cfs instances

Starts or stops a Cloud Factory Services account instance.

### Syntax

- Command Central syntax:

```
cc exec cfs instances accountName instanceId operation [options]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>accountName</i>	Required. Specifies the name of a Cloud Factory Services account.
<i>instanceId</i>	Required. Specifies the ID of the VM instance against which to perform the operation.
<i>operation</i>	Required. Specifies the operation to perform against an instance. Specify one of the following operations: <ul style="list-style-type: none"> <li>■ <code>start</code> - starts an instance</li> </ul>

Argument or Option	Description
	<ul style="list-style-type: none"> <li>■ <code>stop</code> - stops an instance</li> </ul>
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

Instead of specifying the `{--server|-s}` option in the command, you can set the `CC_SERVER` environment variable to the endpoint URL of the Command Central server that hosts Cloud Factory Services as follows:

On Windows: `set CC_SERVER=http://host:port`

On UNIX: `export CC_SERVER="http://host:port/"`

where `host` is the name and `port` is the port of the Command Central server that hosts Cloud Factory Services.

**Note:** The endpoint URL for the Command Central server on which Cloud Factory Services is installed contains *only* host name and port.

For information about `{--server|-s}`, see ["--server | -s" on page 77](#).

### Examples When Executing on Command Central

- To start an instance with ID "i-356ad366" for an Amazon EC2 account with name "ec2bvt" on a Command Central server with host name "rubicon" and port "8490":

```
cc exec cfs instances ec2bvt i-356ad366 start
--server http://rubicon:8490
```

To check if the instance is started and to obtain the run-time status for the instance, use the `cc get cfs instances accountName instanceId {--expected-values|-e} expectedStatus` command.

For example, to obtain the run-time status of an instance with ID "i-356ad366" for an Amazon EC2 account with name "ec2bvt" on a Command Central server with host name "rubicon" and port "8490":

```
cc get cfs instances ec2bvt i-356ad366
--server http://rubicon:8490 -e running
```

For information about `--expected-values|-e`, see ["--expected-values | -e" on page 64](#).

- To stop an instance with ID "i-356ad366" for an Amazon EC2 account with name "ec2bvt" on a Command Central server with host name "rubicon" and port "8490":

```
cc exec cfs instances ec2bvt i-356ad366 stop
--server http://rubicon:8490
```

To check if the instance is stopped and to obtain the run-time status for the instance, use the `cc get cfs instances accountName instanceId [--expected-values|-e] expectedStatus` command.

For example, to obtain the run-time status of an instance with ID “i-356ad366” for an Amazon EC2 account with name “ec2bvt” on a Command Central server with host name “rubicon” and port “8490”:

```
cc get cfs instances ec2bvt i-356ad366
--server http://rubicon:8490 -e stopped
```

For information about `--expected-values|-e`, see “[--expected-values | -e](#)” on page 64.

### Related Commands

[cc create cfs instances](#)

[cc delete cfs instances](#)

[cc get cfs instances](#)

[cc list cfs accounts](#)

[cc list cfs images](#)

[cc list cfs instances](#)

## cc get cfs instances

Finds an instance for a Cloud Factory Services account.

### Syntax

- Command Central syntax:

```
cc get cfs instances accountName instanceId [options]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>accountName</i>	Required. Specifies the name of the Cloud Factory Services account for which you want to find an instance.
<i>instanceId</i>	Required. Specifies the ID of the VM instance for which you want to retrieve information.

Argument or Option	Description
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see " <a href="#">Options for the Commands</a> " on page 59.

### Usage Notes

Instead of specifying the `{--server|-s}` option in the command, you can set the `CC_SERVER` environment variable to the endpoint URL of the Command Central server that hosts Cloud Factory Services as follows:

On Windows: `set CC_SERVER=http://host:port`

On UNIX: `export CC_SERVER="http://host:port/"`

where *host* is the name and *port* is the port of the Command Central server that hosts Cloud Factory Services.

**Note:** The endpoint URL for the Command Central server on which Cloud Factory Services is installed contains *only* host name and port.

For information about `{--server|-s}`, see "[--server | -s](#)" on page 77.

### Example When Executing on Command Central

To find an instance with ID "i-7b40ab2b" for an Amazon EC2 account with name "ec2bvt" on a Command Central server with host name "rubicon" and port "8490":

```
cc get cfs instances ec2bvt i-7b40ab2b --server http://rubicon:8490
```

### Related Commands

[cc create cfs instances](#)

[cc delete cfs instances](#)

[cc exec cfs instances](#)

[cc list cfs accounts](#)

[cc list cfs images](#)

[cc list cfs instances](#)

## cc list cfs accounts

Retrieves a list of all created Cloud Factory Services accounts.

### Syntax

- Command Central syntax:

```
cc list cfs accounts [options]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

Instead of specifying the `{--server|-s}` option in the command, you can set the `CC_SERVER` environment variable to the endpoint URL of the Command Central server that hosts Cloud Factory Services as follows:

On Windows: `set CC_SERVER=http://host:port`

On UNIX: `export CC_SERVER="http://host:port/"`

where *host* is the name and *port* is the port of the Command Central server that hosts Cloud Factory Services.

**Note:** The endpoint URL for the Command Central server on which Cloud Factory Services is installed contains *only* host name and port.

For information about `{--server|-s}`, see ["--server | -s" on page 77](#).

### Examples When Executing on Command Central

To list the registered Cloud Factory Services accounts on a Command Central server with host name "rubicon" and port "8490":

```
cc list cfs accounts --server http://rubicon:8490
```

### Related Commands

[cc create cfs instances](#)

[cc delete cfs instances](#)

[cc exec cfs instances](#)

[cc get cfs instances](#)

[cc list cfs images](#)

[cc list cfs instances](#)

## cc list cfs images

Retrieves a list of the available images for a Cloud Factory Services account.

## Syntax

- Command Central syntax:

```
cc list cfs images accountName [imageId] [options]
```

- Not supported by Platform Manager.

## Arguments and Options

Argument or Option	Description
<i>accountName</i>	Required. Specifies the name of a Cloud Factory Services account.
[ <i>imageId</i> ]	Optional. Specifies the ID of a single image for which you want to retrieve information.
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

## Usage Notes

Instead of specifying the `{--server|-s}` option in the command, you can set the `CC_SERVER` environment variable to the endpoint URL of the Platform Manager server that hosts Cloud Factory Services as follows:

On Windows: `set CC_SERVER=http://host:port`

On UNIX: `export CC_SERVER="http://host:port/"`

where *host* is the name and *port* is the port of the Platform Manager server that hosts Cloud Factory Services.

**Note:** The endpoint URL for the Platform Manager server on which Cloud Factory Services is installed contains *only* host name and port.

For information about `{--server|-s}`, see ["--server | -s" on page 77](#).

## Examples When Executing on Command Central

- To list the available images for an Amazon EC2 account with name "ec2" on a Command Central server with host name "rubicon" and port "8490":

```
cc list cfs images ec2 --server http://rubicon:8490
```

- To find a single image with ID "ami-1f918276" for an Amazon EC2 account with name "ec2" on a Command Central server with host name "rubicon" and port "8490":

```
cc list cfs images ec2 ami-1f918276 --server http://rubicon:8490
```

## Related Commands

[cc create cfs instances](#)

[cc delete cfs instances](#)

[cc exec cfs instances](#)

[cc get cfs instances](#)

[cc list cfs accounts](#)

[cc list cfs instances](#)

## cc list cfs instances

Lists the instances for a Cloud Factory Services account.

### Syntax

- Command Central syntax:

```
cc list cfs instances accountName [instanceId] [options]
```

- Not supported by Platform Manager.

### Arguments and Options

Argument or Option	Description
<i>accountName</i>	Required. Specifies the name of the Cloud Factory Services account for which you want to list the available instances.
[ <i>instanceId</i> ]	Optional. Specifies the ID of a single VM instance for which you want to retrieve information.
[ <i>options</i> ]	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

### Usage Notes

Instead of specifying the `{--server|-s}` option in the command, you can set the `CC_SERVER` environment variable to the endpoint URL of the Command Central server that hosts Cloud Factory Services as follows:

On Windows: `set CC_SERVER=http://host:port`

On UNIX: `export CC_SERVER="http://host:port/"`

where *host* is the name and *port* is the port of the Command Central server that hosts Cloud Factory Services.

**Note:** The endpoint URL for the Command Central server on which Cloud Factory Services is installed contains *only* host name and port.

For information about `{--server|-s}`, see "`--server | -s`" on page 77.

### Examples When Executing on Command Central

- To list the available instances for an Amazon EC2 account with name "ec2bvt" on a Command Central server with host name "rubicon" and port "8490":

```
cc list cfs instances ec2bvt --server http://rubicon:8490
```

- To find a single instance with ID "i-7b40ab2b" for an Amazon EC2 account with name "ec2bvt" on a Command Central server with host name "rubicon" and port "8490":

```
cc list cfs instances ec2bvt i-7b40ab2b --server http://rubicon:8490
```

### Related Commands

[cc create cfs instances](#)

[cc delete cfs instances](#)

[cc exec cfs instances](#)

[cc get cfs instances](#)

[cc list cfs accounts](#)

[cc list cfs images](#)

## cc list cfs vendors

Lists the cloud service providers for which you can create Cloud Factory Services accounts.

### Syntax

- Command Central syntax:

```
cc list cfs vendors [options]
```

- Not supported by Platform Manager.

## Arguments and Options

Argument or Option	Description
<code>[options]</code>	Optional. The command allows all options supported by the Command Line Interface. For a description of the options, see <a href="#">"Options for the Commands" on page 59</a> .

## Usage Notes

Instead of specifying the `{--server|-s}` option in the command, you can set the `CC_SERVER` environment variable to the endpoint URL of the Command Central server that hosts Cloud Factory Services as follows:

On Windows: `set CC_SERVER=http://host:port`

On UNIX: `export CC_SERVER="http://host:port/"`

where *host* is the name and *port* is the port of the Command Central server that hosts Cloud Factory Services.

**Note:** The endpoint URL for the Command Central server on which Cloud Factory Services is installed contains *only* host name and port.

For information about `{--server|-s}`, see ["--server | -s" on page 77](#).

## Examples When Executing on Command Central

To list the cloud service providers for which you can create Cloud Factory Services accounts on a Command Central server with host name "rubicon" and port "8490":

```
cc list cfs vendors --server http://rubicon:8490
```

## Related Commands

[cc create cfs instances](#)

[cc delete cfs instances](#)

[cc exec cfs instances](#)

[cc get cfs instances](#)

[cc list cfs accounts](#)

[cc list cfs images](#)

[cc list cfs instances](#)

---

# 19 Configuring Integration Server

---

■ About Integration Server Configuration Types .....	438
■ Working with Integration Server Configuration Types .....	438
■ Configuring Integration Server Ports .....	439
■ Monitoring KPIs of Integration Server Instances .....	446
■ Integration Server and the Command Line Interface .....	447

---

## About Integration Server Configuration Types

---

You can use the various configuration types that Command Central provides to configure the following settings on Integration Server:

- Clustering
- Database functional aliases
- Email settings
- Global variables
- Integration Server Core and Terracotta license files
- Integration Server loggers and server log facilities
- JDBC connection pools
- JMS settings
- JNDI settings
- Keystore and truststore aliases
- Ports
- Proxy server aliases
- Remote server endpoints
- Resource settings
- Server configuration parameters
- User management
- webMethods Messaging settings
- Web service endpoint alias - Provider
- Web service endpoint alias - Consumer

**Note:** Integration Server must be running if you want to administer Integration Server through Command Central. This section assumes that you are familiar with Integration Server Administrator. For more information about the Integration Server Administrator, see *webMethods Integration Server Administrator's Guide*.

---

## Working with Integration Server Configuration Types

---

Perform the following procedure to add, edit, or delete items for Integration Server configuration type items over Command Central.

**Note:** Ensure that Integration Server is running before performing the following procedure.

### To add, edit, or delete an item for an Integration Server configuration type

1. Select the Integration Server environment from the Environment pane, then click the instance from the **Instances** tab.
2. Click the **Configuration** tab.
3. Select the configuration type from the drop-down list.

Command Central displays the available or default values, if any for the selected Integration Server configuration type.

4. To add an item for the Integration Server configuration type, click . Enter the required values in the displayed fields and click **Save**.

**Note:** For more information about the usage and field descriptions of the Integration Server configuration types, see *webMethods Integration Server Administrator's Guide* or *webMethods Integration Server Online Help*.

5. To edit an item for a configuration type, click on the item that you want to update and click **Edit**. Make the necessary changes and click one of the following:
  - **Test** to test the configuration type item.
  - **Save** to save your changes.
  - **Cancel** to cancel the edits to the configuration type item.
6. To delete an item for a configuration instance, click .

## Configuring Integration Server Ports

### About Ports

You can use Command Central to configure the ports for multiple Integration Servers managed by Software AG Platform Manager. Each port is configured to work with a specific protocol. You can associate an HTTP, HTTPS, FTP, or FTPS with one or more additional ports as needed. By default, Integration Server is pre-configured with HTTP and diagnostic ports at 5555 and 9999, respectively.

**Note:** This section assumes that you are familiar with adding ports in the Integration Server Administrator. For more information about the Integration Server Administrator or Integration Server ports, see *webMethods Integration Server Administrator's Guide*.

You can configure Integration Server ports for HTTP, HTTPS, FTP, and FTPS protocols. In addition, Integration Server supports HTTP and HTTPS *diagnostic ports*. Diagnostic

ports are ports that use threads from a dedicated thread pool to accept requests via HTTP or HTTPS. Diagnostic ports use a dedicated thread pool so that you can access Integration Server when it becomes unresponsive.

**Note:** You can configure only one diagnostic port per Integration Server.

<u>Use this port type...</u>	<u>To...</u>
HTTP	Submit unsecured requests to the server.
HTTPS	Submit requests to the server using SSL encryption.
FTP	Move files to and from the server.
FTPS	Move files to and from the server using SSL encryption.
HTTP Diagnostic	Access Integration Server Administrator when the server becomes unresponsive.
HTTPS Diagnostic	Access Integration Server Administrator using SSL encryption when the server becomes unresponsive.

Before configuring an HTTPS or FTPS port, you must configure Integration Server to use SSL. Use the Integration Server Administrator to create keystore and truststore aliases and certificate mappings. For more information about configuring a port for SSL, see *webMethods Integration Server Administrator's Guide*.

## Configuring Ports

Perform the following procedure to configure Integration Server ports over Command Central.

### To configure ports

1. Select the Integration Server environment from the Environment pane, then click the instance from the **Instances** tab.
2. Click the **Configuration** tab.
3. Select **Ports** in the drop-down list.  
Command Central displays the Integration Server ports.
4. Click  **Add Port**.  
Command Central displays the Select Port Type dialog box.
5. Select one of the following from the **Port Type** drop-down list and click **OK**:

- HTTP
- HTTPS
- FTP
- FTPS
- HTTP Diagnostic
- HTTP Diagnostic

6. Expand **Connection Basics** and complete the following fields:

Field	Specify
<b>Enabled</b>	Whether the port is enabled.
<b>Port Number</b>	The number you want to use for the port. Select a number that is not already in use.
<b>Alias</b>	Name that you want to use for the port alias. Use an alias name that is unique for the instance or component and can be included in a user-friendly URL. The <i>only</i> valid characters in an alias name are ASCII characters, numbers, underscore (_), dot (.), and a hyphen (-).
<b>Bind Address</b>	IP address to which to bind this port. Specify a bind address if your machine has multiple IP addresses and you want the port to use this specific address. If you do not specify a bind address, the server picks one for you.
<b>Backlog</b>	How long a connection request should stay in the queue for a suspended port, before the request is rejected. The default is set to 200 milliseconds (ms), with a maximum permissible value of 65535 ms.
<b>Keep Alive Timeout</b>	When to close the connection if the server has not received a request from the client within this timeout value (in milliseconds); or when to close the connection if the client has explicitly placed a close request with the server.
<b>Package Name</b>	The package associated with this port. When you enable the package, the server enables the port.  When you disable the package, the server disables the port. If you replicate this package, Integration Server creates a port with this number and the same settings on the target server. If a port with this number already exists on the target

Field	Specify
	server, its settings remain intact. This feature is useful if you create an application that expects input on a specific port. The application will continue to work after it is replicated to another server.

7. Expand **Threadpool Configuration** and complete the following fields:

Field	Specify
<b>Enabled</b>	<p>Whether the listener will use this pool exclusively for dispatching requests. The existing thread pool is a global thread pool. If there is a very high load on this resource, the user may have to wait for the global thread pool to process his request. However, with the private thread pool option enabled, requests coming into this port will not have to compete with other server functions for threads.</p> <p>When you view the port's details, the server reports the total number of private thread pool threads currently in use for the port.</p> <p>Click <b>Yes</b> to enable the private thread pool settings. If you do not need to use the thread pool feature, click <b>No</b>.</p>
<b>Threadpool Min</b>	The minimum number of threads for this private thread pool. The default is 1.
<b>Threadpool Max</b>	The maximum number of threads for this private thread pool. The default is 5.
<b>Threadpool Priority</b>	<p>The Java thread priority. The default is 5.</p> <p><b>Important</b> Use this setting with extreme care because it will affect server performance and throughput.</p>

8. If you are creating an HTTPS, HTTPS diagnostic, or FTPS port, expand **Security Configuration** and complete the following fields:

Field	Specify
<b>Client Authentication</b>	<p>The type of client authentication you want Integration Server to perform for requests that arrive on the port. Select:</p> <ul style="list-style-type: none"> <li>■ <b>Username/Password</b> if you want to use basic authentication.</li> </ul>

Field	Specify
	<ul style="list-style-type: none"> <li data-bbox="524 323 1365 464">■ <b>REQUEST_CERTIFICATE</b> if you want Integration Server to request client certificates for all requests. If the client does not provide a certificate, the server prompts the client for a userid and password. If the client provides a certificate: <ul style="list-style-type: none"> <li data-bbox="570 485 1365 688">■ The server checks whether the certificate exactly matches a client certificate on file and is signed by a trusted authority. If so, the client is logged in as the user to which the certificate is mapped in Integration Server. If not, the client request fails, unless central user management is configured.</li> <li data-bbox="570 709 1365 850">■ If central user management is configured, the server checks whether the certificate is mapped to a user in the central user database. If so, the server logs the client on as that user. If not, the client request fails.</li> </ul> </li> <li data-bbox="524 871 1365 1003">■ <b>REQUIRE_CERTIFICATE</b> if you want Integration Server to require client certificates for all requests. The server behaves as described for <b>REQUEST_CERTIFICATE</b>, except that the client must always provide a certificate.</li> </ul>
<b>Keystore Alias</b>	<p data-bbox="524 1045 1289 1108">Optional. A user-specified, text identifier for an Integration Server keystore.</p> <p data-bbox="524 1129 1289 1304">The alias points to a repository of private keys and their associated certificates. Although each listener points to one keystore, there can be multiple keys and their certificates in the same keystore, and more than one listener can use the same keystore alias.</p>
<b>Key Alias</b>	<p data-bbox="524 1346 1305 1415">Optional. The alias for the private key, which must be stored in the keystore specified by the above keystore alias.</p>
<b>Truststore Alias</b>	<p data-bbox="524 1457 1305 1623">Optional. The alias for the truststore. The truststore must contain the trusted root certificate for the CA that signed the Integration Server certificate associated with the key alias. The truststore also contains the list of CA certificates that Integration Server uses to validate the trust relationship.</p>

9. Expand **IP Access Restrictions** and specify the following to allow or deny access from specified ports:

Field	Specifies
<b>Use Global Default</b>	That the port should use the global IP access settings set in Integration Server. This is the default.
<b>Allow by Default</b>	That the port should allow requests from all hosts except for ones you explicitly deny. This setting overrides Integration Server's global IP access setting for this port. Use this approach if you want to allow most hosts and deny a few.
<b>Deny by Default</b>	That the port should deny requests from all hosts except for ones you explicitly allow. This setting overrides Integration Server's global IP access setting for this port. Use this approach if you want to deny most hosts and allow a few.
<b>Hosts to allow</b>	<p>The host names (example, workstation5.webmethods.com) or IP addresses (example, 132.906.19.22 or 2001:db8:85a3:8d3:1319:8a2e:370:7348) of hosts from which the server is to accept inbound requests. Enter each host name on a separate line.</p> <p>The host names or IP addresses can include upper and lower case alphabetic characters, digits (0-9), hyphens (-), and periods (.) but cannot include spaces. For IPv6, IP addresses can also include colons (:) and brackets ([]).</p>

10. Expand **URL Access Restrictions** and specify the following to allow or deny access to specified service URLs:

Field	Specifies
<b>Use Global Default</b>	That the port should use the default mode access settings set in Integration Server. This is the default.
<b>Deny by Default</b>	That the port should deny requests from all service URLs except for ones you explicitly allow. Use this approach if you want to deny most URLs and allow a few.
<b>Allow by Default</b>	That the port should allow requests from all service URLs except for ones you explicitly deny. Use this approach if you want to deny most hosts and allow a few.
<b>URLs to deny</b>	The service URLs from which Integration Server is to accept inbound requests. Enter each service URL on a separate line.

11. Click **Test** to test the port.
12. Click **Save**.

## Testing Ports

Perform the following procedure to test Integration Server ports.

---

### To test ports

1. Select the Integration Server environment from the Environment pane, then click the instance from the **Instances** tab.
2. Click the **Configuration** tab.
3. Select **Ports** in the drop-down list.
4. From the **Port** column, locate the port you want to test, and click on the port number.
5. Click **Test**.

## Setting the Primary Port

Perform the following procedure to set the Integration Server primary port.

---

### To set the primary port

1. Select the Integration Server environment from the Environment pane, then click the instance from the **Instances** tab.
2. Click the **Configuration** tab.
3. Select **Ports** in the drop-down list.
4. Locate the port you want to designate as the primary port and click on the port number.
5. Click  and then **Set as Primary**.
6. Click **Ok** on the confirmation prompt.

## Editing Port Information

Perform the following procedure to edit port information.

**Note:** You cannot change an existing port alias.

---

### To edit port information

1. Select the Integration Server environment from the Environment pane, then click the instance from the **Instances** tab.

2. Click the **Configuration** tab.
3. Select **Ports** in the drop-down list.
4. From the **Port** column, locate the port whose details you want to edit, and click on the port number.
5. Click **Edit**.
6. Make changes to the port and click one of the following:
  - **Test** to test the port.
  - **Save** to change your edits to the port.
  - **Cancel** to cancel the edits to the port.

## Enabling and Disabling Ports

Perform the following procedure to enable or disable a port.

---

### To enable or disable a port

1. Select the Integration Server environment from the Environment pane, then click the instance from the **Instances** tab.
2. Click the **Configuration** tab.
3. Select **Ports** in the drop-down list.
4. From the **Port** column, locate the port you want to enable or disable, and click on the port number.
  - The  icon indicates that the port is enabled.
  - The  icon indicates that the port is disabled.
5. Expand **Threadpool Configuration** and enable or disable the port.

## Monitoring KPIs of Integration Server Instances

Perform the following procedure to monitor KPIs of Integration Server instances.

---

### To view the KPIs of Integration Server instances

1. On the **Environments** pane, select the environment you want to monitor.
2. Click the **Instances** tab.
3. In the table, select the Integration Server you want to monitor.
4. Click the **Overview** tab.

The **Monitoring** section in the **Dashboard** shows the KPIs of the Integration Server instance.

Integration Server returns the following three KPIs:

Name	Marginal Value	Critical Value	Maximum Value
Average response time (in ms)	80% of maximum	95% of maximum	5000
Service errors	70% of maximum	90% of maximum	5
Running services	80% of maximum	95% of maximum	10

## Integration Server and the Command Line Interface

### Command Central Command Line Tool Upgrade

In all existing Integration Server automation scripts and queries that use the Command Central REST API or command line tool commands, change the Integration Server instance ID for all Integration Server instances from:

```
runtimeComponent=integrationServer-ENGINE
```

to

```
runtimeComponent=integrationServer-instanceName
```

where *instanceName* is the name of the Integration Server instance, for example `integrationServer-default`.

### Commands that Integration Server Supports

Integration Server supports the Platform Manager commands listed in the following table. The table lists where you can find general information about each command. Additionally, if there is Integration Server-specific information, the table lists where you can learn more about arguments and options that Integration Server supports or details about the actions Integration Server takes when you execute an `exec` command.

Commands	Additional Information
<code>cc create configuration data</code>	<p>For general information about the command, see <a href="#">"cc create configuration data" on page 247</a>.</p> <p>For Integration Server-specific information about using this command, see</p>

Commands	Additional Information
	<p><a href="#">"Configuration Types that IntegrationServer-instanceName Supports"</a> on page 450.</p>
<p>cc delete configuration data</p>	<p>For general information about the command, see <a href="#">"cc delete configuration data"</a> on page 251.</p>
<p>cc get configuration data</p>	<p>For general information about the command, see <a href="#">"cc get configuration data"</a> on page 253.</p>
<p>cc update configuration data</p>	<p>For general information about the command, see <a href="#">"cc update configuration data"</a> on page 256.</p>
<p>cc get configuration instances</p>	<p>For general information about the command, see <a href="#">"cc get configuration instances"</a> on page 260.</p> <p>For Integration Server-specific information about using this command, see <a href="#">"Configuration Types that IntegrationServer-instanceName Supports"</a> on page 450.</p>
<p>cc list configuration instances</p>	<p>For general information about the command, see <a href="#">"cc list configuration instances"</a> on page 263.</p> <p>For Integration Server-specific information about using this command, see <a href="#">"Configuration Types that IntegrationServer-instanceName Supports"</a> on page 450.</p>
<p>cc get configuration types</p>	<p>For general information about the command, see <a href="#">"cc get configuration types"</a> on page 265.</p> <p>For Integration Server-specific information about using this command, see <a href="#">"Configuration Types that IntegrationServer-instanceName Supports"</a> on page 450.</p>
<p>cc list configuration types</p>	<p>For general information about the command, see <a href="#">"cc list configuration types"</a> on page 267.</p> <p>For Integration Server-specific information about using this command, see <a href="#">"Configuration Types that IntegrationServer-instanceName Supports"</a> on page 450.</p>

Commands	Additional Information
cc exec configuration validation create	<p>For general information about the command, see <a href="#">"cc create configuration data" on page 247</a>.</p> <p>For Integration Server-specific information about using this command, see <a href="#">"Configuration Types that IntegrationServer-instanceName Supports" on page 450</a>.</p>
cc exec configuration validation delete	<p>For general information about the command, see <a href="#">"cc create configuration data" on page 247</a>.</p>
cc exec configuration validation update	<p>For general information about the command, see <a href="#">"cc create configuration data" on page 247</a>.</p> <p>For Integration Server-specific information about using this command, see <a href="#">"Configuration Types that IntegrationServer-instanceName Supports" on page 450</a>.</p>
cc create instances	<p>For general information about the command, see <a href="#">"cc create instances" on page 277</a>.</p>
cc delete instances	<p>For general information about the command, see <a href="#">"cc delete instances" on page 279</a>.</p> <div data-bbox="743 1176 1373 1276" style="background-color: #f0f0f0; padding: 5px;"> <p><b>Important:</b> You must stop an Integration Server instance before deleting the instance.</p> </div>
cc list instances supported products	<p>For general information about the command, see <a href="#">"cc list instances supported products" on page 282</a>.</p>
cc update instances	<p>For general information about the command, see <a href="#">"cc update instances" on page 283</a>.</p>
cc get inventory components	<p>For general information about the command, see <a href="#">"cc get inventory components" on page 194</a>.</p>
cc list inventory components	<p>For general information about the command, see <a href="#">"cc list inventory components" on page 198</a>.</p>

Commands	Additional Information
<code>cc exec lifecycle</code>	For general information about the command, see <a href="#">"cc exec lifecycle" on page 236</a> .
<code>cc get monitoring</code>	<p>For general information about the command, see <a href="#">"cc get monitoring" on page 227</a>.</p> <p>For Integration Server-specific information about using this command, see:</p> <ul style="list-style-type: none"> <li>■ <a href="#">"Run-time Monitoring Statuses for IntegrationServer-instanceName " on page 456</a></li> <li>■ <a href="#">"Run-time Monitoring States for IntegrationServer-instanceName " on page 457</a></li> </ul>

## Configuration Types that IntegrationServer-*instanceName* Supports

IntegrationServer-*instanceName*, where *instanceName* is the name of the Integration Server instance, run-time component supports creating instances of the configuration types listed in the following table.

Configuration Type	Use to configure...
COMMON-ADMINUI	Full URL to Integration Server.
COMMON-CLUSTER	Configuration instance to enable or disable a cluster configuration and to specify cluster settings such as the Terracotta Server array URLs and the session timeout.
COMMON-DBFUNCTION	Configuration instance for database functional aliases for Integration Server.
COMMON-JDBC	Configuration instance for JDBC connection pools for Integration Server.
COMMON-JMS	Configuration instance for JMS settings for Integration Server.
COMMON-JNDI	Configuration instance for JNDI settings for Integration Server.

Configuration Type	Use to configure...
COMMON-KEYSTORES	Configuration instance for a KeyStore alias that identifies a keystore file or private key within a keystore.
COMMON-LICENSE	License files. <code>IntegrationServer-instanceName</code> supports configuration instances for the Integration Server Core and Terracotta license files.
COMMON-LICLOC	<p>Locations where license files reside in file system where Integration Server is installed. <code>IntegrationServer-instanceName</code> supports configuration instances for the location of the Integration Server Core license file and for the location of the Terracotta license file.</p> <p>You cannot change the location of the license file.</p>
COMMON-LOCAL-USERS	<p>Configuration instance to manage local users and their passwords. <code>COMMON-LOCAL-USERS-&lt;userId&gt;</code> supports configuring the various users. Each user must be declared using a separate configuration instance.</p> <p>By default, there are four predefined <code>COMMON-LOCAL-USERS</code> configuration instances, namely, <code>COMMON-LOCAL-USERS-Administrator</code>, <code>COMMON-LOCAL-USERS-Default</code>, <code>COMMON-LOCAL-USERS-Developer</code>, and <code>COMMON-LOCAL-USERS-Replicator</code>.</p> <p><b>Note:</b> Upon creation, a user is automatically added to the Everybody group.</p>
COMMON-LOGGERS	<p>Configuration instance for Integration Server loggers and server log facilities.</p> <p>For Server Logger, the default value for Levels of Loggers is <code>Inherit</code>, which indicates that the components will have the same logging levels as its parent. You cannot add or delete Logger Levels and can only modify the Level values.</p>

Configuration Type	Use to configure...
COMMON-PORTS	Ports. IntegrationServer- <i>instanceName</i> supports configuration instances for HTTP, HTTPS, FTP, FTPS, and Diagnostics ports.
COMMON-PROXY	Configuration instance for proxy server aliases.  <b>Note:</b> You can use the default proxy alias, <b>proxy-bypass</b> , to optionally route selected requests directly to their targets, bypassing the proxy. The proxy bypass addresses list applies to all proxy server aliases. You cannot create or delete the <b>proxy-bypass</b> alias.
COMMON-SMTP	Settings for sending e-mail messages.
COMMON-SYSPROPS	Server configuration parameters. IntegrationServer- <i>instanceName</i> supports viewing and updating the system configuration parameters defined in the server.cnf configuration file:  <i>Integration Server_directory/instances/instance_name/config/server.cnf</i> , where <i>instance_name</i> is the name of the Integration Server instance.
COMMON-TRUSTSTORES	Configuration instance for a TrustStore alias that identifies a truststore file.
COMMON-VARS	Configuration instance for global variables. Each variable must be declared using a separate configuration instance.
COMMON-WMMESSAGING	Configuration instance for webMethods Messaging settings for Integration Server. IntegrationServer- <i>instanceName</i> supports configuration instances for webMethods Messaging providers (Broker and Universal Messaging).
IS-CONSUMER-ENDPOINTS	Consumer web service endpoint aliases.

Configuration Type	Use to configure...
IS-DEFAULT-WMMESSAGING	The default messaging connection alias.
IS-FILEPERMISSION	File access control configuration for the <code>pub.file</code> services in <code>WmPublic</code> package. The <code>IS-FILEPERMISSION</code> configuration instance includes <code>IS-FILEPERMISSION_READ</code> , <code>IS-FILEPERMISSION_WRITE</code> , and <code>IS-FILEPERMISSION_DELETE</code> . You cannot create a new configuration instance or delete an existing configuration instance. For more information about file access control configuration, see <i>webMethods Integration Server Built-In Services Reference</i> .
IS-PRIMARYPORT	The primary port ID for Integration Server.
IS-PROVIDER-ENDPOINTS	Provider web service endpoint aliases.
IS-QUIESCEPORT	The quiesce port ID for Integration Server.
IS-RESOURCES	Configuration instance for resource settings for Integration Server. <code>IntegrationServer-instanceName</code> supports configuration instances for outbound HTTP, stateful sessions limit, server thread pool, document stores, and XA recovery store settings.

### Related Commands

[cc create configuration data](#)

[cc get configuration instances](#)

[cc list configuration instances](#)

[cc get configuration types](#)

[cc list configuration types](#)

## Lifecycle Actions for Integration Server

The following table lists the actions that Integration Server supports with the `cc exec lifecycle` command and the operation taken against Integration Server when an action is executed.

Action	Description
pause	Switches an active Integration Server to quiesce mode. The Integration Server run-time status is set to PAUSED. When an Integration Server is paused, all ports except the diagnostic port and the quiesce port are disabled. In quiesce mode, any requests that are already in progress are permitted to finish, but any new inbound requests to the server are blocked. Outbound connection attempts, such as connections to JDBC pools or connections through LDAP or a central user directory, remain open.
resume	Switches an Integration Server in quiesce mode to active mode and resumes normal operation. All the assets and activities that were disabled or suspended are restored or resumed. The Integration Server run-time status returns to ONLINE.

## Integration Server Instance Management

The following table lists the required parameters that you must include when managing Integration Server instances using the Command Central instance management commands:

Command	Parameter	Description
<code>cc create instances</code>	<code>instance.name=name</code>	Required. A name for the new Integration Server instance.
	<code>primary.port=port</code>	Required. The main listening port for the new Integration Server instance.
	<code>diagnostic.port=port</code>	Required. The diagnostic port for the new Integration Server instance.
	<code>jmx.port=port</code>	Optional. The JMX port for the new Integration Server instance.
	<code>install.service=false   true</code>	Optional. Specifies whether to install Integration Server as an

Command	Parameter	Description
		<p>application or a service. Valid values:</p> <ul style="list-style-type: none"> <li>■ <code>true</code> - install as a service</li> <li>■ <code>false</code> - install as an application (default)</li> </ul>
	<code>license.file=location_of_license_file</code>	<p><b>Optional.</b> The location of your Integration Server license file.</p> <p>The script copies the license file from the specified location to <code>IntegrationServer/instances/instance_name/config/licenseKey.xml</code> file.</p> <p>This parameter is optional; however, if a license file location is not specified, the Integration Server instance will shut down after 30 minutes.</p> <p><b>Note:</b> the location name contains spaces, you must surround the name of the location with quotation marks (" ").</p>
	<code>db.alias=database_alias_name</code>	<p><b>Optional.</b> The alias name for the database. The default value is <code>JDBC_POOL_ALIAS</code>.</p>
	<code>db.type=database_type</code>	<p><b>Required.</b> The type of database used by the new server instance. Specify one of the following databases:</p> <ul style="list-style-type: none"> <li>■ <code>sql</code> - Microsoft SQL Server</li> <li>■ <code>oracle</code> - Oracle</li> <li>■ <code>db2</code> - DB2</li> </ul> <p>These values are not case-sensitive. The default is the embedded IS internal database.</p>

Command	Parameter	Description
	<code>db.username=database_username</code>	Required. The user name assigned to the Integration Server database.
	<code>db.password=database_password</code>	Required. The password of the Integration Server database.
	<code>db.url=database_URL</code>	Required. The connection URL for the database.

### Example When Executing on Command Central

- To create the new instance for an installed Integration Server with instance name “is-instance2”, diagnostic port “8083”, and primary port “8081” in the installation with alias name “productionNode2”:

```
cc create instances productionNode2 integrationServer
instance.name=is-instance2 diagnostic.port=8083
primary.port=8081
db.alias=sql_server db.type=SQL
db.username=sa db.password=password@1234
db.url="jdbc:wm:sqlserver://10.60.72.87:1433;databaseName=SQL_DB"
```

### Examples When Executing on Platform Manager

- To create the new instance for an installed Integration Server with instance name “is-instance2”, diagnostic port “8083”, JMX port “10058”, and primary port “8081”:

```
cc create instances integrationServer instance.name=is-instance2
diagnostic.port=8083 jmx.port=10058 primary.port=8081
db.alias=sql_server db.type=SQL
db.username=sa db.password=password@1234
db.url="jdbc:wm:sqlserver://10.60.72.87:1433;databaseName=SQL_DB"
```

- To create the new instance for an installed Integration Server using the instance data in the `instance-settings.properties` file, located in the current directory:

```
cc create instances integrationServer -i instance-settings.properties
```

- To create the new instance for an installed Integration Server using the instance data in the `instance.settings.xml` file, located in the current directory:

```
cc create instances integrationServer -i instance-settings.xml
```

### Related Commands

[cc create instances](#)

## Run-time Monitoring Statuses for IntegrationServer-*instanceName*

The following table lists the run-time statuses that the IntegrationServer-*instanceName* run-time component can return in response to the `cc get monitoring runtimestatus`

and `cc get monitoring state` commands, along with the meaning of each run-time status.

Run-time Status	Meaning
ONLINE	The Integration Server is running, and it is accepting and processing requests over the Integration Server primary port.
PAUSED	The Integration Server is in quiesce mode. Integration Server is accepting or processing requests only over the diagnostic port and the quiesce port.
STOPPED	The Integration Server has been stopped. Integration Server is not accepting nor processing requests over the Integration Server primary port.
UNKNOWN	The status of the Integration Server cannot be determined.
UNRESPONSIVE	The Integration Server is running, but not reachable.

**Note:** `IS-instanceName` might still report ONLINE, which indicates there is an issue with Integration Server.

### Related Commands

[cc get monitoring](#)

## Run-time Monitoring States for `IntegrationServer-instanceName`

In response to the `cc get monitoring runtimestate` and `cc get monitoring state` commands, `IntegrationServer-instanceName` run-time component provides information about the following key performance indicators (KPIs):

KPI	Description
Running Services	<p>Use this KPI to monitor the number of services that Integration Server is running concurrently so that you can take corrective actions if the number approaches a critical value. The number of running services includes services that were triggered, scheduled, or directly invoked.</p> <p>The KPI uses the following marginal, critical, and maximum values:</p>

KPI	Description
	<ul style="list-style-type: none"> <li>■ Marginal is 80% of the high water mark of concurrently running services.</li> <li>■ Critical is 95% of the high water mark of concurrently running services.</li> <li>■ Maximum is 100% of the high water mark of concurrently running services. This is shown in the Threads area of Integration Server.</li> </ul>
Response Time	<p>Use this KPI to monitor service response time so that you can take corrective actions if the response time approaches a critical value.</p> <p>The KPI uses the following marginal, critical, and maximum values:</p> <ul style="list-style-type: none"> <li>■ Marginal is 80% of the high water mark of service response time.</li> <li>■ Critical is 95% of the high water mark of service response time.</li> <li>■ Maximum is 100% of the high water mark of service response time.</li> </ul>
Service Errors	<p>Use this KPI to monitor how many service exceptions have occurred in the last minute so that you can take corrective actions if the current number of exceptions is approaching a critical value.</p> <p>The KPI uses the following marginal, critical, and maximum values:</p> <ul style="list-style-type: none"> <li>■ Marginal is 5 exceptions in the last minute.</li> <li>■ Critical is 20 exceptions in the last minute.</li> <li>■ Maximum is more than 20 exceptions in the last minute.</li> </ul>

### Related Commands

[cc get monitoring](#)

---

# 20

## Configuring My webMethods Server

---

■	Configuring My webMethods Server Ports .....	460
■	Editing Port Settings .....	460
■	Configuring My webMethods Server Email .....	461
■	Working with My webMethods Server Environment Variables .....	462
■	Monitoring KPIs of My webMethods Server Instances .....	463
■	Using Trusted Authentication to Connect to My webMethods Server .....	464
■	My webMethods Server and the Command Line Interface .....	464

## Configuring My webMethods Server Ports

---

My webMethods Server listens for client requests on one or more ports. When a port receives a message or request, My webMethods Server invokes the appropriate services. Each port is configured to work with a specific protocol. You can associate HTTP or HTTPS with one or more additional ports as needed. By default, My webMethods Server is pre-configured with HTTP at 8585.

The MWS\_default component is the OSGi profile. The My webMethods Server component is the standard profile for the server instance. You can edit configuration settings for the My webMethods Server component, but you cannot add or delete them.

---

### To configure My webMethods Server ports

1. In the Environments pane, select the environment in which you want to view the My webMethods Server instance.
2. Select the **Instances** tab.
3. Expand the *MWS\_mwsinstancename* node containing the My webMethods Server instance you want to configure.
4. Click **My webMethods Server** in the name column.
5. Select the **Configuration** tab. Make sure **My webMethods Server** is selected in the left pane.
6. Select **Ports** from the drop-down list box.
7. **Test** and **Save** the port.

## Editing Port Settings

---

Perform the following procedure to change the port settings.

---

### To enable or disable a port

1. Select the My webMethods Server environment from the Environments pane, then click the My webMethods Server instance you want to edit from the **Instances** tab.
2. Click the **Configuration** tab.
3. Select **Ports** in the drop-down list.
4. Click the number of the port you want to edit and click **Edit**. The port settings are now editable. Make the necessary changes to the port settings.
5. **Test** and **Save** the changes.

## Configuring My webMethods Server Email

Perform the following procedure to configure My webMethods Server email.

### To configure email

1. Select the My webMethods Server environment from the Environment pane, then click the instance from the **Instances** tab.
2. Click the **Configuration** tab.
3. Select **Email** in the drop-down list.  
Command Central displays the My webMethods Server SMTP Server Configuration.
4. Click **Edit**.
5. In **Connection Basics**, complete the following fields.

Field	Specify
<b>Server Name</b>	The SMTP server's host name. For example: smtp.server.com.
<b>Port</b>	The SMTP server's port number.
<b>Sender Name</b>	The default name to use in the <b>From</b> field of the email messages sent by the server.
<b>Sender Email</b>	The default email address to use in the <b>From</b> field of the email messages sent by the server.

6. Expand **Advanced Settings** and complete the following fields.

Field	Description
<b>SMTP Username</b>	Optional. The user name that My webMethods Server has to supply for authentication. If the SMTP server requires authentication, specify the user name.
<b>SMTP Password</b>	Optional. The password associated with the <b>SMTP Username</b> . If the SMTP server requires authentication, specify the appropriate password.

7. Click **Test** and **Save** the email settings.

## Working with My webMethods Server Environment Variables

### Considerations when Configuring My webMethods Server Variables

You should consider the following naming conventions when configuring My webMethods Server environment variables:

- The length of the variable name and value must not exceed 255 characters.
- Variable names are case-sensitive.
- No restrictions for special characters apply to the variable name.

### Configuring My webMethods Server Variables

When adding a My webMethods Server environment variable in the Command Central user interface, select one of the following Custom types:

- **Default.** Creates variables without secure fields.
- **Secure.** Creates password variables whose values are encrypted by My webMethods Server.

When adding or editing My webMethods Server global environment variables in the Command Central user interface, specify values for the following fields:

Field	Description
<b>Key</b>	Required. The name of the My webMethods Server global environment variable. My webMethods Server uses the key to refer to the environment variable while performing environment variable substitution.  <b>Note:</b> The <b>Key</b> field is disabled when editing a variable.
<b>Type</b>	Required. The data type of the My webMethods Server global environment variable.  <b>Note:</b> The <b>Type</b> field is disabled when editing a variable.
<b>Value</b>	Required. The value of the My webMethods Server global environment variable.  When you add or edit a secure environment variable, the value you specify in this field is hidden. After you save the environment variable, the <b>Value</b> field remains empty.

**Important:** After you edit or delete an environment variable, the webMethods applications using the variable are automatically restarted.

For detailed steps about how to add, edit, or delete a global environment variable, see "[Configuring Instances](#)" on page 242.

## Monitoring KPIs of My webMethods Server Instances

### To view the KPIs of My webMethods Server instances

1. On the Environments pane, select the environment you want to monitor.
2. Click the **Instances** tab.
3. Select the My webMethods Server you want to monitor.
4. Click the **Overview** tab.

The **Monitoring** section in the **Dashboard** shows the KPIs of the My webMethods Server instance.

My webMethods Server returns the following three KPIs.

Name	Marginal Value	Critical Value	Maximum Value
<b>Number of user sessions</b>	80% of maximum	95% of maximum	At least 100, or high water mark.  (High water mark is the highest value ever reached.)
<b>JDBC connection pool size (maximum number of connections to JDBC)</b>	80% of maximum	95% of maximum	As configured.
<b>Average response time (in milliseconds)</b>	50% of maximum	90% of maximum	At least 10 seconds, or high water mark.

---

## Using Trusted Authentication to Connect to My webMethods Server

---

By default Command Central version 9.8 and higher uses SAML-based trusted authentication when connecting to My webMethods Server. You do not need to perform any operation to enable this functionality.

However, if you change the value of the **Assertion Parameter Name** field in the My webMethods Server administration web user interface, you must configure an additional Java parameter, `MWS.SAML.paramname`, in the `custom_wrapper.conf` file of the Platform Manager that manages the My webMethods Server run-time component. The `MWS.SAML.paramname` parameter specifies the changed Assertion Parameter Name value. For more information about setting the Assertion Parameter Name value, see *Administering My webMethods Server*.

---

### To configure the `MWS.SAML.paramname` parameter in Platform Manager

1. Go to the `Software AG_directory \profiles\SPM\configuration` directory and open the `custom_wrapper.conf` file in a text editor.
2. Add the following parameter:

```
wrapper.java.additional.number =  
-DMWS.SAML.paramname=assertionParameterName
```

where *number* is a unique sequential number depending on the already existing parameter numbers in the file, and *assertionParameterName* is the modified value of the **Assertion Parameter Name** field in the My webMethods Server administration user interface.

3. Save the file and restart Platform Manager.

---

## My webMethods Server and the Command Line Interface

---

### Commands that My webMethods Server Supports

My webMethods Server supports the Platform Manager commands listed in the following table. The table lists where you can find general information about each command. Additionally, the table lists where you can learn more about arguments and options that My webMethods Server supports or details about the actions My webMethods Server takes when you execute an `exec` command.

Commands	For more information, see...
<code>cc create configuration data</code>	<p>For general information about the command, see <a href="#">"cc create configuration data"</a> on page 247.</p> <p>For My webMethods Server-specific information about using this command, see <a href="#">"Configuration Types that My webMethods Server-ENGINE Supports"</a> on page 467.</p>
<code>cc delete configuration data</code>	<p>For general information about the command, see <a href="#">"cc delete configuration data"</a> on page 251.</p>
<code>cc get configuration data</code>	<p>For general information about the command, see <a href="#">"cc get configuration data"</a> on page 253.</p> <p>For My webMethods Server-specific information about using this command, see <a href="#">"Configuration Types that My webMethods Server-ENGINE Supports"</a> on page 467.</p>
<code>cc update configuration data</code>	<p>For general information about the command, see <a href="#">"cc update configuration data"</a> on page 256.</p>
<code>cc get configuration instances</code>	<p>For general information about the command, see <a href="#">"cc get configuration instances"</a> on page 260.</p>
<code>cc list configuration instances</code>	<p>For general information about the command, see <a href="#">"cc list configuration instances"</a> on page 263.</p> <p>For My webMethods Server-specific information about using this command, see <a href="#">"Configuration Types that My webMethods Server-ENGINE Supports"</a> on page 467.</p>
<code>cc get configuration types</code>	<p>For general information about the command, see <a href="#">"cc get configuration types"</a> on page 265.</p> <p>For My webMethods Server-specific information about using this command, see <a href="#">"Configuration Types that My webMethods Server-ENGINE Supports"</a> on page 467.</p>

Commands	For more information, see...
cc list configuration types	<p>For general information about the command, see <a href="#">"cc list configuration types" on page 267.</a></p> <p>For My webMethods Server-specific information about using this command, see <a href="#">"Configuration Types that My webMethods Server-ENGINE Supports" on page 467.</a></p>
cc exec configuration validation create	<p>For general information about the command, see <a href="#">"cc create configuration data" on page 247.</a></p> <p>For My webMethods Server-specific information about using this command, see <a href="#">"Configuration Types that My webMethods Server-ENGINE Supports" on page 467.</a></p>
cc exec configuration validation delete	<p>For general information about the command, see <a href="#">"cc create configuration data" on page 247.</a></p>
cc exec configuration validation update	<p>For general information about the command, see <a href="#">"cc create configuration data" on page 247.</a></p> <p>For My webMethods Server-specific information about using this command, see <a href="#">"Configuration Types that My webMethods Server-ENGINE Supports" on page 467.</a></p>
cc create instances	<p>For general information about the command, see <a href="#">"cc create instances" on page 277.</a></p> <p>For My webMethods Server-specific information about using the command, see <a href="#">"Creating a My webMethods Server Instance" on page 470.</a></p>
cc delete instances	<p>For general information about the command, see <a href="#">"cc delete instances" on page 279.</a></p>
cc list instances supported products	<p>For general information about the command, see <a href="#">"cc list instances supported products" on page 282.</a></p>
cc update instances	<p>For general information about the command, see <a href="#">"cc update instances" on page 283.</a></p>

Commands	For more information, see...
	For My webMethods Server-specific information about using the command, see <a href="#">"Updating a My webMethods Server Instance" on page 474.</a>
cc get inventory components	For general information about the command, see <a href="#">"cc get inventory components" on page 194.</a>
cc list inventory components	For general information about the command, see <a href="#">"cc list inventory components" on page 198.</a>
cc update inventory components	For general information about the command, see <a href="#">"cc update inventory components" on page 205.</a>
cc exec lifecycle	<p>For general information about the command, see <a href="#">"cc exec lifecycle" on page 236.</a></p> <p>For My webMethods Server-specific information about using this command, see <a href="#">"Lifecycle Actions for My webMethods Server-ENGINE" on page 470.</a></p>
cc get monitoring	<p>For general information about the command, see <a href="#">"cc get monitoring" on page 227.</a></p> <p>For My webMethods Server-specific information about using this command, see:</p> <ul style="list-style-type: none"> <li>■ <a href="#">"Run-time Monitoring Statuses for My webMethods Server-ENGINE" on page 475</a></li> <li>■ <a href="#">"Run-time Monitoring States for My webMethods Server " on page 476</a></li> </ul>

## Configuration Types that My webMethods Server-ENGINE Supports

My webMethods Server-ENGINE run-time component supports creating instances of the configuration types listed in the following table.

Configuration Type	Use to configure...
COMMON-ADMINUI	<p>Full URL to My webMethods Server.</p> <ul style="list-style-type: none"> <li>■ If the My webMethods Server HTTP port is enabled, use the following format where <i>hostname</i> is the My webMethods Server host name and <i>httpport</i> is the My webMethods Server HTTP port number. <code>http://hostname :httpport</code></li> <li>■ If the My webMethods Server HTTPS port is enabled and the HTTP port is <i>not</i> enabled, use the following format where <i>hostname</i> is the My webMethods Server host name and <i>httpsport</i> is the My webMethods Server HTTPS port number. <code>https://hostname :httpsport</code></li> </ul>
COMMON-CLUSTER	<p>Settings for a My webMethods Server cluster. You can configure the load balancer, JMS provider, and node roles for a My webMethods Server cluster. For information about the fields and the values to specify, see <i>Administering My webMethods Server</i>.</p> <p><b>Important:</b> Any changes to the cluster configuration take effect only after you restart the node.</p>
COMMON-JDBC	<p>The default connection pool for the My webMethods Server database connection. You can use the command line interface to edit the pool, but not delete it.</p> <p>You can also add, update, or delete additional custom JDBC pools that custom Composite Application Framework (CAF) applications running on My webMethods Server use.</p> <p><b>Note:</b> You can manage instances of this configuration type using the command line interface, but not the Command Central user interface.</p>
COMMON-KEYSTORES	<p>Keystores for My webMethods Server. You can edit the keystores that My webMethods Server uses for its HTTPS port to provide your own keystores.</p>

Configuration Type	Use to configure...
	<p><b>Note:</b> You can manage instances of this configuration type using the command line interface, but not the Command Central user interface.</p>
COMMON-LOCAL-USERS	<p>Settings for managing internal users of a My webMethods Server instance. COMMON-LOCAL-USERS-<i>userId</i> supports configuring the details for each user. For information about the fields and values to specify, see <i>Administering My webMethods Server</i>.</p>
COMMON-PORTS	<p>The My webMethods Server HTTP and HTTPS ports.</p> <p>When adding, editing, and removing ports, keep the following in mind:</p> <ul style="list-style-type: none"> <li>■ Ensure at least an HTTP or an HTTPS port is defined.</li> <li>■ You can only delete the HTTPS port if the HTTP port is defined. If you delete the HTTPS port, you can later add it again.</li> <li>■ You can only delete the HTTP port if the HTTPS port is defined. If you delete the HTTP port, you can later add it again.</li> <li>■ There are no restrictions with regards to editing port numbers.</li> </ul> <p><b>Note:</b> You can also manage ports in the Command Central user interface if the ports are enabled. After enabling the ports in My webMethods Server Cluster Administration and restarting My webMethods Server, the ports are visible in the Command Central user interface.</p>
COMMON-SMTP	<p>Settings for sending e-mail messages.</p>
COMMON-TRUSTORES	<p>Truststores for My webMethods Server. You can edit the truststores that My webMethods Server uses for its HTTPS port to provide your own truststores.</p>

Configuration Type	Use to configure...
	<b>Note:</b> You can manage instances of this configuration type using the command line interface, but not the Command Central user interface.
COMMON-VARS	Global environment variables for My webMethods Server.

### Related Commands and Topics

[cc create configuration data](#)

[cc get configuration instances](#)

[cc list configuration instances](#)

[cc get configuration types](#)

[cc list configuration types](#)

## Lifecycle Actions for My webMethods Server-ENGINE

The My webMethods Server-ENGINE run-time component does not support the use of the [cc exec lifecycle](#) command to perform lifecycle actions.

### Related Commands

[cc exec lifecycle](#)

## My webMethods Server Instance Management

### Creating a My webMethods Server Instance

The following table lists the parameters to include when creating a My webMethods Server instance using the Command Central instance management commands.

Command	Parameter	Description
<a href="#">cc create instances</a>	<code>MwsProgramFiles</code>	Required. The product ID of the installed My webMethods Server.
	<code>instance.name=<i>name</i></code>	Required. The name of the new My webMethods Server instance. The default name is <code>default</code> .

Command	Parameter	Description
	<code>db.type={sqlserver ms oracle db2}</code>	<p>Required. The type of database used by the new server instance. Specify one of the following databases:</p> <ul style="list-style-type: none"> <li>■ <code>sqlserver</code> or <code>ms</code> - Microsoft SQL Server</li> <li>■ <code>oracle</code> - Oracle</li> <li>■ <code>db2</code> - DB2</li> </ul>
	<code>db.url=url</code>	Required. The connection URL for the database.
	<code>db.username=name</code>	Required. The user name assigned to the My webMethods Server database.
	<code>db.password=password</code>	Required. The password of the My webMethods Server database user.
	<code>db.driver=name</code>	Optional. The class name of the JDBC driver used to connect to the My webMethods Server database.
	<code>node.name=name</code>	Optional. The name of the cluster node that will host the new server instance. The default name is <code>master</code> .
	<code>http.port=port</code>	Optional. The port number on which the new server instance listens. The default port number is <code>8585</code> .
	<code>https.port=port</code>	Optional. The https port on which the new server instance listens. A value of <code>0</code> disables the listener.
	<code>debug.port=port</code>	Optional. The Java debug port for the new server instance. The default port number is <code>10033</code> .
	<code>jmx.port=port</code>	Optional. The JMX port for the new server instance. The default port number is <code>5002</code> .

Command	Parameter	Description
	<code>server.features={core default fabric all}</code>	<p>Optional. The set of component features configured with the new server instance. Specify one of the following values:</p> <ul style="list-style-type: none"> <li>■ <code>core</code> - The minimum set of features needed to support development of JSR 168 portlets using Software AG Designer. This value indicates a pure runtime, with a single skin and shell, and no administration or configuration features.</li> <li>■ <code>default</code> - The standard set of My webMethods Server features, but without the additional features or development tools found in the <i>Software AG_directory\MWS\components</i> directory. This is the default value.</li> <li>■ <code>fabric</code> - The default My webMethods Server taxonomy with all installed My webMethods Server user interfaces.</li> <li>■ <code>all</code> - The standard set of features plus all components found in the <i>Software AG_directory\MWS\components</i> directory.</li> </ul>
	<code>components.include=name</code>	Optional. Components in the <i>Software AG_directory\MWS\components</i> directory to include in the new server instance.
	<code>components.exclude=name</code>	Optional. Components in the <i>Software AG_directory\MWS\components</i> directory to exclude from the new server instance.
	<code>components.override={true false}</code>	Optional. Specifies whether to overwrite the component files in the <i>Software AG_directory\MWS\server\server_name\deploy</i> directory that are older than the component files in the <i>Software AG_directory\MWS\components</i> directory. Valid values are:

Command	Parameter	Description
		<ul style="list-style-type: none"> <li>■ true - overwrite component files in the \deploy directory.</li> <li>■ false - do not overwrite component files in the \deploy directory. This is the default value.</li> </ul>

### Examples When Executing on Command Central

- To create a new instance for an installed My webMethods Server with instance name "test1", database type "sqlserver", database user name "mws", database password "mws", and database URL "jdbc:wm:sqlserver://localhost:1433;databaseName=testDB" in the installation with alias name "local":

```
cc create instances local MwsProgramFiles instance.name=test1
db.type=sqlserver db.username=mws db.password=mws
db.url=jdbc:wm:sqlserver://localhost:1433;databaseName=testDB
```

- To create a new instance for an installed My webMethods Server with instance name "test1", database type "sqlserver", database user name "mws", database password "mws", database URL "jdbc:wm:sqlserver://localhost:1433;databaseName=testDB", database driver "com.wm.dd.jdbc.sqlserver.SQLServerDriver", http port "9595", and JMX port "7000" in the installation with alias name "local":

```
cc create instances local MwsProgramFiles instance.name=test1
db.type=sqlserver db.username=mws db.password=mws
db.url=jdbc:wm:sqlserver://localhost:1433;databaseName=testDB
db.driver=com.wm.dd.jdbc.sqlserver.SQLServerDriver http.port=9595
jmx.port=7000
```

### Examples When Executing on Platform Manager

- To create a new instance for an installed My webMethods Server with instance name "test1", database type "sqlserver", database user name "mws", database password "mws", and database URL "jdbc:wm:sqlserver://localhost:1433;databaseName=testDB":

```
cc create instances MwsProgramFiles instance.name=test1
db.type=sqlserver db.username=mws db.password=mws
db.url=jdbc:wm:sqlserver://localhost:1433;databaseName=testDB
```

- To create a new instance for an installed My webMethods Server with instance name "test1", database type "sqlserver", database user name "mws", database password "mws", database URL "jdbc:wm:sqlserver://localhost:1433;databaseName=testDB", database driver "com.wm.dd.jdbc.sqlserver.SQLServerDriver", http port "9595", and JMX port "7000":

```
cc create instances MwsProgramFiles instance.name=test1
db.type=sqlserver db.username=mws db.password=mws
db.url=jdbc:wm:sqlserver://localhost:1433;databaseName=testDB
db.driver=com.wm.dd.jdbc.sqlserver.SQLServerDriver http.port=9595
jmx.port=7000
```

## Related Commands

[cc create instances](#)

## Updating a My webMethods Server Instance

The following table lists the parameters to include when updating a My webMethods Server instance using the Command Central instance management commands.

Command	Parameter	Description
<a href="#">cc update instances</a>	<code>MwsProgramFiles- instance.name</code>	Required. The ID of the My webMethods Server instance that you want to update. For example: <code>MwsProgramFiles-test1</code> .
	<code>components.override= {true false}</code>	Optional. Specifies whether to overwrite the component files in the <i>Software AG_directory\MWS\server\server_name\deploy</i> directory that are older than the components files in the <i>Software AG_directory\MWS\components</i> directory. Valid values are: <ul style="list-style-type: none"> <li>■ <code>true</code> - overwrite component files in the <code>\deploy</code> directory.</li> <li>■ <code>false</code> - do not overwrite component files in the <code>\deploy</code> directory. This is the default value.</li> </ul>
	<code>dont.update.classpath= {true false}</code>	Optional. Specifies whether to update the generated My webMethods Server classpath. Specify one of the following values <ul style="list-style-type: none"> <li>■ <code>true</code> - do not update the server classpath.</li> <li>■ <code>false</code> - update the server classpath. This is the default value.</li> </ul>
	<code>platformdir=path</code>	Optional. The path to the platform installation directory. The default path is <i>Software AG_directory</i> .

Command	Parameter	Description
	<code>timeout=number</code>	Optional. Specifies how many seconds to wait for the command to complete before terminating the attempt to execute the command.

For more information about how to update a My webMethods Server instance, see "[cc update instances](#)" on page 283.

### Related Commands

[cc update instances](#)

## Deleting a My webMethods Server Instance

You can delete a My webMethods Server instance using the Command Central command line interface. For more information about how to delete a My webMethods Server instance, see "[cc delete instances](#)" on page 279.

## Run-time Monitoring Statuses for My webMethods Server-ENGINE

The following table lists the run-time statuses that the My webMethods Server-ENGINE run-time component can return in response to the `cc get monitoring runtimestatus` and `cc get monitoring state` commands, along with the meaning of each run-time status.

Run-time Status	Meaning
ONLINE	<p>The My webMethods Server-ENGINE run-time component is running.</p> <p>The run-time component indicates ONLINE when the profile JVM is running and that the My webMethods Server port is responding.</p>
FAILED	<p>The My webMethods Server-ENGINE run-time component is not running due to some failure, and attempts to start it again have failed.</p> <p>The run-time component indicates FAILED when the profile JVM is no longer running, but the Tanuki wrapper PID file still exists.</p>
STARTING	<p>The My webMethods Server-ENGINE run-time component is starting.</p>

Run-time Status	Meaning
	The run-time component indicates STARTING when the server is starting and reports an HTTP 503, "Not Ready", status from its HTTP port.
STOPPING	The My webMethods Server-ENGINE run-time component is stopping.
UNKNOWN	The status of The My webMethods Server-ENGINE run-time component cannot be determined.
UNRESPONSIVE	The My webMethods Server-ENGINE run-time component does not respond to a ping to its HTTP/S port.

### Related Commands

[cc get monitoring](#)

## Run-time Monitoring States for My webMethods Server

In response to the `cc get monitoring runtimestate` and `cc get monitoring state` commands, My webMethods Server provides information about the following key performance indicators (KPIs):

KPI	Description
Number of user sessions	<p>Use this KPI to monitor the number of active user sessions so that you can take corrective actions if the number approaches a critical value.</p> <ul style="list-style-type: none"> <li>■ Marginal is 80 active user sessions.</li> <li>■ Critical is 95 active user sessions.</li> <li>■ Maximum is 100 or more active user sessions.</li> </ul>
Number of active connections in the JDBC pool	<p>Use this KPI to monitor the number of active connections in the JDBC pool so that you can take corrective actions if the number of connections approaches a critical value.</p> <ul style="list-style-type: none"> <li>■ Marginal is 90 active connections.</li> <li>■ Critical is 95 active connections.</li> <li>■ Maximum is 100 or more active connections.</li> </ul>

KPI	Description
Average response times in milliseconds	<p data-bbox="589 317 1383 426">Use this KPI to monitor My webMethods Server response times so that you can take corrective actions if the response times slow to a critical value.</p> <ul data-bbox="589 443 1383 623" style="list-style-type: none"><li data-bbox="589 443 1383 485">■ Marginal is 5000 milliseconds.</li><li data-bbox="589 495 1383 537">■ Critical is 9000 milliseconds.</li><li data-bbox="589 548 1383 623">■ Maximum is response times at 10000 or more millisecond.</li></ul>

---

**Related Commands**

[cc get monitoring](#)



---

# 21 Configuring OSGi Profiles

---

■ Protocols that Command Central Supports in OSGi Profiles .....	480
■ Default Ports .....	481
■ Products that Support Port Configuration in OSGi Profiles .....	481
■ Port Authentication .....	482
■ Configuring Ports in OSGi Profiles .....	483
■ OSGi Components and the Command Line Interface .....	488

Command Central uses the ports specified in the OSGi profiles of products for monitoring the managed products. You can add, modify, or delete the ports in the OSGi profiles.

Command Central and Platform Manager listen for requests on ports that you specify. Each port is associated with a protocol, such as HTTP or HTTPS. In addition to these ports, Command Central uses JMX ports for alerts.

## Protocols that Command Central Supports in OSGi Profiles

Command Central supports HTTP, HTTPS, JMX, SSH, and JDWP. JMX, SSH, and JDWP protocols allow only one port each.

Use this port type...	To...
HTTP	Submit unsecured requests to the OSGi component.
HTTPS	Submit requests to the OSGi component using SSL encryption.
JMX	<p>Allow administration and monitoring the JVM KPIs of the OSGi component.</p> <p>To monitor the product-specific KPI's of the Integration Server, My webMethods Server, and Platform Manager instances, you need not enable the JMX port in the OSGi profile of the corresponding product.</p> <p>To view the inventory, run-time status (enabled/disabled), and to start/stop (or enable/disable) the Integration Server packages, do the following:</p> <ul style="list-style-type: none"> <li>■ Enable the JMX port in the OSGi profile of Integration Server.</li> <li>■ Enable <code>subsystem</code> in the manifest file of the Integration Server package as shown below. <pre>&lt;Values version="2.0"&gt; &lt;value name="subsystem"&gt;true&lt;/value&gt; &lt;/Values&gt;</pre> </li> </ul> <p>JMX port might be bound to a localhost.</p>
SSH	Allow secure shell for the OSGi component.
JDWP	Allow OSGi component debugging by using the Java debug protocol over a TCP connection.

To enable the HTTP/HTTPS ports of Integration Server, configure the HTTP/HTTPS ports of the Integration Server instance, not the ports in the OSGi profile.

To enable the HTTP/HTTPS ports of My webMethods Server, configure the HTTP/HTTPS ports of the My webMethods Server instance, not the ports in the OSGi profile.

## Default Ports

When you install Command Central and Platform Manager, Software AG Installer assigns the HTTP and HTTPS port numbers. If the default port numbers are used by other products, Installer displays available ports. You can select one of the available ports, or you can change the port manually.

The following table shows the default HTTP and HTTPS ports.

Product	Alias	Port Type	Default Port Assignment
Command Central	defaultHttp	HTTP	8090
	defaultHttps	HTTPS	8091
Platform Manager	defaultHttp	HTTP	8092
	defaultHttps	HTTPS	8093

As an administrator, you can change the default port assignments by modifying the configuration settings using the Command Central web user interface or the command line interface. For commands and options, see *Software AG Command Central and Software AG Platform Manager Command Reference*.

## Products that Support Port Configuration in OSGi Profiles

The following table lists the products that have OSGi profiles that support port configuration.

Product	Product Code	Ports Enabled by Default
Command Central	CCE	JMX, HTTP, HTTPS
Integration Server	IS	JMX
My webMethods Server	MWS_ <i>mwsinstancename</i>	JMX

Product	Product Code	Ports Enabled by Default
	For example, MWS_default	
Platform Manager	SPM	JMX, HTTP, HTTPS
Software AG Runtime	CTP	JMX, HTTP, HTTPS

All these products support SSH and JDWP port configuration. For information about configuring the ports in the OSGi profile of a product, see [Configuring Ports in OSGi Profiles](#).

**Note:** Integration Server and My webMethods Server have two profiles: OSGi profile and the instance profile. For more information about configuring an Integration Server instance, see "[Configuring Integration Server Ports](#)" on page 439. For more information about configuring a My webMethods Server instance, see "[Configuring My webMethods Server](#)" on page 459 and "[Configuring My webMethods Server Email](#)" on page 461.

## Port Authentication

The following table describes which user store products with OSGi profiles use to authenticate enabled ports.

Product	Product Code	Authenticates against user store in...	For...
Command Central	CCE	<i>Software AG_directory</i> \profiles\CCE \configuration\security\users.txt	All ports
Platform Manager	SPM	<i>Software AG_directory</i> \profiles \SPM\configuration\security \users.txt	All ports
Software AG Runtime	CTP	<i>Software AG_directory</i> \common \conf\users.txt	All ports
Infrastructure Data Collector	InfraDC	<i>Software AG_directory</i> \common \conf\users.txt	All ports
My webMethods Server	MWS_ <i>instancename</i>  For example, MWS_default	<i>Software AG_directory</i> \common \conf\users.txt	JMX and SSH

Product	Product Code	Authenticates against user store in...	For...
		user store managed by My webMethods Server	HTTP, HTTPS
Integration Server	IS_ <i>instancename</i> For example, IS_default	user store managed by Integration Server	All ports

Integration Server can open JMX port using a setting in the `com.software.jmx.connector.pid-port.properties` file located in the `Software AG_directory\profiles\IS_instance_name\configuration\com.softwareag.platform.config.propsloader` directory, where *instance\_name* is the name of the Integration Server instance. For more information about enabling JMX monitoring in Integration Server, see the *webMethods Integration Server Administrator's Guide*.

## Configuring Ports in OSGi Profiles

You configure ports in the Configuration tab of a product-specific OSGi instance.

### Adding Ports

Perform the following procedure to configure new ports in the OSGi profiles.

#### To add a port

1. In the Environments pane, in the **Instances** tab, click the OSGi instance or component to which you want to add a port.
2. Click the **Configuration** tab.
3. Click the  to add a new port.
4. Select one of the following in **Port Type** and click **OK**:
  - HTTP
  - HTTPS
  - JMX
  - SSH
  - JDWP
5. In Connection Basics, configure the fields corresponding to the port type.
  - For HTTP and HTTPS port configurations:

---

<b>Field</b>	<b>Description</b>
<b>Enabled</b>	Whether the port is enabled.
<b>Port Number</b>	The number you want to use for the port. Select a number that is not already in use.
<b>Alias</b>	Name that you want to use for the port alias. Use an alias name that is unique for the instance or component and can be included in a user-friendly URL. The <i>only</i> valid characters in an alias name are ASCII characters, numbers, underscore (_), dot (.), and a hyphen (-).
<b>Keep Alive Timeout</b>	When to close the connection if the server has not received a request from the client within this timeout value (in milliseconds); or when to close the connection if the client has explicitly placed a close request with the server.
<b>Spare Threads Min</b>	The starting number of request processing spare threads.
<b>Redirect Port</b>	The port to use when redirecting a SSL connection requests.
<b>Spare Threads Max</b>	The maximum number of request processing spare threads.
<b>Accept Count</b>	The maximum number of simultaneous connection requests allowed in the connection queue.
<b>Connection Timeout</b>	The connection timeout in milliseconds. This attribute is not set by default on HTTPS ports.
<b>HTTP Header Size Max</b>	The maximum incoming URL length in characters.
<b>Upload Timeout Disable</b>	Indicates if using a longer connection timeout is allowed when waiting for the servlet container to update. <ul style="list-style-type: none"><li>■ <b>Yes.</b> Allow longer connection time-outs while waiting for the servlet container.</li></ul>

Field	Description
	<ul style="list-style-type: none"> <li>■ <b>No.</b> Do not allow longer connection time-outs.</li> </ul>
<b>Lookups Enable</b>	<p>Indicates if DNS lookups are allowed to get the actual host name of a remote client.</p> <ul style="list-style-type: none"> <li>■ <b>Yes.</b> DNS lookups allowed.</li> <li>■ <b>No.</b> DNS lookups not allowed.</li> </ul>
<b>Key Manager Algorithm</b>	For HTTPS port configurations. The certificate encoding algorithm.
<b>SSL Protocol</b>	For HTTPS port configurations. The version of the secure socket layer (SSL) protocol to use; when not specified Transport Layer Security ((TLS) is used.
<ul style="list-style-type: none"> <li>■ For JMX and SSH port configurations:</li> </ul>	
Field	Description
<b>Enabled</b>	Whether the port is enabled.
<b>Port Number</b>	<p>The number you want to use for the port. Select a number that is not already in use.</p> <ul style="list-style-type: none"> <li>■ For JMX, select the port for monitoring, managing, and implementing the Java process.</li> <li>■ For SSH, select the port for remote shell services or execution processes.</li> </ul>
<b>Alias</b>	<p>Name that you want to use for the port alias. Use an alias name that is unique for the instance or component and can be included in a user-friendly URL. The <i>only</i> valid characters in an alias name are ASCII characters, numbers, underscore (_), dot (.), and a hyphen (-).</p>
<b>JAAS Realm</b>	For JMX and SSH port configurations. Specifies the realm name that authenticates the Java Authentication and Authorization (JAAS) service.
<ul style="list-style-type: none"> <li>■ For JDWP port configurations:</li> </ul>	
<b>Note:</b>	The JDWP port is only used when the profile is started in debug mode.

Field	Description
<b>Port Number</b>	The number you want to use for the port. Select a number that is not already in use.
<b>Alias</b>	Name that you want to use for the port alias. Use an alias name that is unique for the instance or component and can be included in a user-friendly URL. The <i>only</i> valid characters in an alias name are ASCII characters, numbers, underscore (_), dot (.), and a hyphen (-).
<b>Suspend</b>	For JDWP port configurations. Select Yes if the runtime should be suspended until debugger connects.

6. In Thread pool Configuration, for HTTP and HTTPS ports, complete the following fields.

Field	Description
<b>Enabled</b>	<p>Whether the listener uses this pool exclusively for dispatching requests. The existing thread pool is a global thread pool. If there is a very high load on this resource, there may be a delay until the global thread pool can process the request. However, with the private thread pool option enabled, requests coming into this port do not compete with other server functions for threads.</p> <p>When you view the port's details, the server reports the total number of private thread pool threads currently in use for the port.</p> <p>Click <b>Yes</b> to enable the private thread pool settings. If you do not need to use the thread pool feature, click <b>No</b>.</p>
<b>Threadpool Min</b>	The minimum number of threads for this private thread pool. The default is 1.
<b>Threadpool Max</b>	The maximum number of threads for this private thread pool. The default is 5.
<b>Threadpool priority</b>	<p>The Java thread priority. The default is 5.</p> <p><b>Important</b> Use this setting with extreme care because it will affect server performance and throughput.</p>

7. For secure connections, complete the security fields as follows:

<b>Field</b>	<b>Description</b>
<b>SSL Enabled</b>	Whether secure layering is enabled.  Click <b>Yes</b> to enable the private thread pool settings. If you do not need to use the thread pool feature, click <b>No</b> .
<b>Keystore Type</b>	Select the keystore type. The keystore must contain the private key for secure communication.
<b>Server Location of Keystore</b>	Specify the directory where the keystore file is located.
<b>Password</b>	Specify the password to open the keystore file.
<b>Truststore Type</b>	Select the truststore type. The truststore must contain the trusted root certificate for the CA that signed the OSGi component certificate associated with the key alias. The truststore also contains the list of CA certificates that OSGi component uses to validate the trust relationship.
<b>Server Location of Truststore</b>	Specify the directory where the truststore file is located.
<b>Password</b>	Specify the password to open the truststore file.

## Viewing the Port Settings

Use the following procedure to view the settings for an existing port.

### To view the platform manager ports

1. In the Environments pane, click the environment in which you want to view the OSGi instance or component.
2. Click the **Instances** tab.
3. Click the name of the OSGi instance or component you want to view.
4. Click the **Configuration** tab.
5. In **Ports**, select the port. The field displays the parameters available for configuration.

---

## Editing and Testing OSGi Port Information

Perform the following procedure to edit OSGi port information.

**Note:** You cannot change an existing port alias.

---

### To edit port information

1. In the Environments pane, click the environment in which you want to edit the OSGi profile.instance from the **Instances** tab.
2. Click the **Instances** tab.
3. Click the name of the OSGi instance or component you want to view.
4. Click the **Configuration** tab.
5. In **Ports**, select the port. The field displays the parameters available for configuration.
6. Locate the port whose details you want to edit, and click on the port number.
7. Click **Edit**.
8. Make changes to the port and click **Test** or **Save**.

## Deleting a Port

Use the following procedure to delete a port configuration from an OSGi profile.

---

### To delete a port

1. In the Environments pane, click the environment in which you want to view the OSGi instance.
2. Click the **Instances** tab.
3. Click the name of the OSGi instance or component.
4. Click the **Configuration** tab.
5. Select the port that you want to delete and click .

**Note:** You can only delete ports that are disabled.

---

## OSGi Components and the Command Line Interface

### Configuration Types that the OSGi Profile Components Supports

The OSGi profile run-time components support creating instances of one or more of the following configuration types:

- COMMON-JAAS
- COMMON-LOCAL-USERS
- COMMON-LOG
- COMMON-MEMORY
- COMMON-PORTS
- COMMON-PROXY
- COMMON-SYSPROPS

The following sections provide more detail about each configuration type and list the products whose OSGi profile supports each configuration type.

### COMMON-JAAS

Use to configure profile authentication and authorization configuration:

*installation\_directory* /profiles/*product\_code* /configuration/jaas.config file

For Command Central and Platform Manager, the jaas.config file is located under:

*installation\_directory* /profiles/*product\_code* /configuration/security/jaas.config file

**Note:** For the *product\_code* in the directory path, refer to the description of COMMON-SYSPROPS.

#### Products whose OSGi profile supports COMMON-JAAS

Command Central

Platform Manager

Software AG Runtime

For more information, see JAAS file information in the ["Using JAAS with Command Central"](#) on page 180.

### COMMON-LOCAL-USERS

Use to manage internal users for a product instance.

#### Products whose OSGi profile supports COMMON-LOCAL-USERS

Command Central

Integration Server

My webMethods Server

---

**Products whose OSGi profile supports COMMON-LOCAL-USERS**


---

Platform Manager

---

 Universal Messaging
 

---

**COMMON-LOG**

Use to configure log levels for log categories and log file locations.

---

**Products whose OSGi profile supports COMMON-LOG**


---

Command Central

---

 Platform Manager
 

---

Software AG Runtime

**COMMON-MEMORY**

Use to configure common memory settings.

---

**Products whose OSGi profile supports COMMON-MEMORY**


---

Command Central

---

 Platform Manager
 

---

Software AG Runtime

**COMMON-PORTS**

Use to configure the HTTP, HTTPS, JMX, JDWP (Debug), and/or SSH ports. The following tables lists products that have OSGi profile components that support COMMON-PORTS and the types of ports each support.

Port type	Products whose OSGi profile supports this port type	Description
HTTP	Command Central Platform Manager Software AG Runtime	The HTTP port is enabled by default. You can use the command line interface to add, remove, edit, and validate the HTTP port.

---

Port type	Products whose OSGi profile supports this port type	Description
HTTPS	Command Central Platform Manager Software AG Runtime	The HTTPS port is enabled by default. You can use the command line interface to add, remove, edit, and validate the HTTPS port.
JMX	Command Central Integration Server My webMethods Server Platform Manager Software AG Runtime	The JMX port is enabled by default. You can use the command line interface to add, remove, and edit the JMX port. You can only define one JMX port.
JDWP (Debug)	Command Central Integration Server My webMethods Server Platform Manager Software AG Runtime	The JDWP (Debug) port is disabled by default. You can use the command line interface to edit this port, but not to remove or add it. This port is used when the run-time component is started in debug mode using the <code>cc exec lifecycle</code> command.
SSH	Command Central Integration Server My webMethods Server Platform Manager Software AG Runtime	The SSH port is disabled by default. You can use the command line interface to add, remove, and edit the SSH port. You can only define one SSH port.

### COMMON-PROXY

Use to configure proxy server settings if you must route server requests through a third party server. Based on the transport protocol, COMMON-PROXY has the following configuration types:

Configuration types	Description
COMMON-PROXY-HTTP	Use to configure proxy settings over the HTTP protocol.
COMMON-PROXY-HTTPS	Use to configure proxy settings over the HTTPS protocol.

Configuration types	Description
COMMON-PROXY-FTP	Use to configure proxy settings over the FTP protocol.
COMMON-PROXY-SOCKS	Use to configure proxy settings over the SOCKS protocol.
COMMON-PROXY-ALL	Use to indicate whether to use the operating system proxy settings instead of COMMON-PROXY-* configuration.

**Important:** You cannot edit or delete this configuration type.

The following table lists the run-time components whose OSGi profile supports the COMMON-PROXY configuration type:

Products whose OSGi profile supports COMMON-PROXY
Command Central
Platform Manager
My webMethods Server
Software AG Runtime

**Note:** Integration Server supports COMMON-PROXY configuration type at the ENGINE level.

You can use the Command Central web user interface to add, view, edit, and delete a proxy server configuration. For information about how to work with Command Central configuration types using the Command Central web user interface, see ["Configuring Instances" on page 242](#).

You can also create, get, update and delete proxy server settings using the configuration commands in the Command Central command line interface. When creating or updating a proxy server configuration instance, the input XML file that contains the proxy server configuration data must use the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<Proxy alias="HTTPS">
  <Enabled>true</Enabled>
  <Protocol>HTTPS</Protocol>
  <Host>hostName</Host>
  <NonProxyHosts>host1,host2</NonProxyHosts >
  <Port>12321</Port >
  <Username>user</Username >
```

```
<Password>secure</Password >
</Proxy>
```

The following table describes the values you provide for each of the parameters in the proxy server configuration data XML file:

For this parameter...	Specify...
<b>Alias</b>	The alias name to use for this host/port combination. The Alias and Protocol parameters should have the same value.
<b>Enabled</b>	<p>Whether to enable the proxy server configuration instance. By default, the existing proxy server configurations except COMMON-PROXY-ALL are enabled. Valid values are:</p> <ul style="list-style-type: none"> <li>■ <code>true</code> Enabled</li> <li>■ <code>false</code> Disabled</li> </ul> <p>The COMMON-PROXY-ALL configuration type is not enabled by default. For the COMMON-PROXY-ALL configuration type the Enabled parameter indicates whether to use the proxy settings of the operating system and ignore any other COMMON-PROXY-* configurations.</p>
<b>Protocol</b>	The type of protocol to use for the host/port combination. The Alias and Protocol parameters should have the same value.
<b>Host</b>	The host name or IP address of the proxy server. For the COMMON-PROXY-ALL configuration type, you can also specify <code>noHost</code> .
<b>NonProxyHosts</b>	You can optionally route selected requests directly to their targets, bypassing the proxy. To specify non-proxy hosts, type the fully qualified host and domain name of each server that should receive requests directly. To enter multiple names, separate each with commas.
<b>Port</b>	The port on which this proxy server listens for requests. Specify a valid port number in the range of [1-65535].
<b>Username</b>	The user name to use to access this proxy server.
<b>Password</b>	The password to use to access this proxy server.

All proxy server configuration settings are stored in the config.ini configuration file under the OSGi run-time component profile directory. You *must not* modify or edit the config.ini configuration file.

### COMMON-SYSPROPS

Use to configure the OSGi profile properties defined in the config.ini configuration file:

*installation\_directory* /profiles/*product\_code* /configuration/config.ini file

**Note:** Under normal circumstances, you *must not* modify the config.ini file. It should *only* be modified by the Software AG Installer.

The following tables lists products that have OSGi profile components that support COMMON-SYSPROPS and the *product\_code* used in the directory path to the location of the config.ini for that product.

Products whose OSGi profile supports COMMON-SYSPROP	<i>product_code</i> used in the directory path to the config.ini file
Command Central	CCE
Integration Server	IS_ <i>instancename</i> where <i>instancename</i> is the name of the Integration Server instance, for example, "IS_default".
My webMethods Server	MWS_ <i>instancename</i> where <i>instancename</i> is the name of the My webMethods Server instance, for example, "MWS_default".
Platform Manager	SPM
Software AG Runtime	CTP

### Related Commands

[cc create configuration data](#)

[cc get configuration instances](#)

[cc list configuration instances](#)

[cc get configuration types](#)

[cc list configuration types](#)

## Lifecycle Actions for the OSGi Profile Components

The following table lists the actions that the OSGi profile run-time components support with the `cc exec lifecycle` command and the operations taken against a run-time component when an action is executed.

Action	Products whose OSGi component supports this action	Description
start	Command Central Integration Server My webMethods Server Software AG Runtime	Starts the run-time component. When successful, the run-time status is set to ONLINE.  When the run-time component starts, the OSGi framework comes online and opens the JMX port.  <b>Note</b> To correctly report the ONLINE status, the JMX port must be enabled.  <b>Note</b> When using the command line interface, to start Command Central you must execute the command against Platform Manager because the command will fail against Command Central when it is not running.
startindebugmode	Command Central Integration Server My webMethods Server Software AG Runtime	Starts the run-time component in debug mode. When successful, the run-time status is set to ONLINE.
stop	Command Central Integration Server My webMethods Server Software AG Runtime	Stops the run-time component. When successful, the run-time status is set to STOPPED.

Action	Products whose OSGi component supports this action	Description
restart	Command Central Integration Server My webMethods Server Software AG Runtime	Stops, then restarts the run-time component. The run-time status is set to ONLINE.  <b>Note</b> To correctly report the ONLINE status, the JMX port must be enabled.

### Related Commands

[cc exec lifecycle](#)

## Run-time Monitoring Statuses for the OSGi Profile Components

The following table lists the run-time statuses that the OSGi profile run-time components can return in response to the `cc get monitoring runtimestatus` and `cc get monitoring state` commands, along with the meaning of each run-time status.

Run-time Status	Meaning
ONLINE	The run-time component is running.  The run-time component indicates ONLINE when the profile JVM is running and that the JMX port is responding.
FAILED	The run-time component is not running due to some failure, and attempts to start it again have failed.
STARTING	The run-time component is starting.
STOPPED	The run-time component is not running because it was shut down normally.
STOPPING	The run-time component is stopping.
UNKNOWN	The status of the run-time component cannot be determined.
UNRESPONSIVE	The run-time component does not respond to a ping to its JMX port.

## Related Commands

[cc get monitoring](#)

## Run-time Monitoring States for OSGi Profile Components

In response to the `cc get monitoring runtimestate` and `cc get monitoring state` commands, the OSGi profile run-time components provide information about the key performance indicators (KPIs) in the following table.

KPI	Description
JVM memory usage	<p>Use this KPI to monitor the JVM memory usage of the run-time component so that you can take corrective actions if storage use approaches a critical value.</p> <p>The KPI uses the following marginal, critical, and maximum values:</p> <ul style="list-style-type: none"> <li>■ Marginal is calculated using the following:  <math>\text{MAX}(\text{Maximum} * 80\%, \text{Maximum} - 100)</math>            This means a marginal value is when there is only 20% free JVM memory available or less than 100MB of JVM memory is available.</li> <li>■ Critical is calculated using the following:  <math>\text{MAX}(\text{Maximum} * 95\%, \text{Maximum} - 50)</math>            This means a critical value is when there is only 5% free JVM memory available or less than 50MB of JVM memory left.</li> <li>■ Maximum amount of memory that is allocated memory for the JVM.</li> </ul> <p><b>Note:</b> This KPI value might be incorrect when running in a 32-bit operating systems.</p>
Number of JVM threads	<p>Use this KPI to monitor number of JVM threads that the E run-time component is using so that you can take corrective actions if the number of used threads approaches a critical value.</p> <p>The KPI uses the following marginal, critical, and maximum values:</p> <ul style="list-style-type: none"> <li>■ Marginal is 80% of the allocated JVM threads.</li> </ul>

KPI	Description
	<ul style="list-style-type: none"> <li>■ Critical is 95% of the allocated JVM threads.</li> <li>■ Maximum is calculated using the following: MAX(HWM(value), 500)</li> </ul> <p>This means the initial maximum value is 500 threads. However, if the JVM has more than 500 threads, the greater number is used as the maximum.</p>
JVM CPU load	<p>Use this KPI to monitor how much CPU the JVM is using so that you can take corrective actions if the CPU usage approaches a critical value.</p> <p>The KPI uses the following marginal, critical, and maximum values:</p> <ul style="list-style-type: none"> <li>■ Marginal is 80% of the CPU usage.</li> <li>■ Critical is 95% of the CPU usage.</li> <li>■ Maximum is 100% of the CPU usage.</li> </ul> <div style="background-color: #f0f0f0; padding: 5px;"> <p><b>Note:</b> This KPI is <i>only</i> supported when running on Java 7. It is <i>not</i> supported on Java 6.</p> <p><b>Note:</b> This KPI is <i>not</i> reported when running on HP-UX.</p> </div>

### Related Commands

[cc get monitoring](#)

## Configuring the Command Central OSGi Profile Component

See the following sections:

- ["Configuration Types that OSGI-CCE-ENGINE Supports" on page 498](#)
- ["Lifecycle Actions for OSGI-CCE-ENGINE" on page 499](#)
- ["Run-time Monitoring Statuses for OSGI-CCE-ENGINE" on page 499](#)

### Configuration Types that OSGI-CCE-ENGINE Supports

The OSGI-CCE-ENGINE run-time component supports creating instances of the configuration types listed in the following table.

Configuration Type	Use to...
COMMON-LICENSE	Update the Command Central license file.
COMMON-LICLOC	Retrieve the location of the Command Central license file. Updating the license file is not supported.

### Related Commands

#### Configuration Types that the OSGi Profile Components Supports

[cc create configuration data](#)

[cc get configuration instances](#)

[cc list configuration instances](#)

[cc get configuration types](#)

[cc list configuration types](#)

### Lifecycle Actions for OSGI-CCE-ENGINE

The following table lists the actions that the OSGI-CCE-ENGINE run-time component supports with the [cc exec lifecycle](#) command and the operation taken against the run-time component when an action is executed.

Action	Description
restart	Stops, then restarts the run-time component. When successful, the run-time status is set to ONLINE.

### Related Commands

[cc exec lifecycle](#)

### Run-time Monitoring Statuses for OSGI-CCE-ENGINE

The following table lists the run-time statuses that the OSGI-CCE-ENGINE run-time component can return in response to the [cc get monitoring runtimestatus](#) and [cc get monitoring state](#) commands, along with the meaning of each run-time status.

Run-time Status	Meaning
ONLINE	The run-time component is running.

Run-time Status	Meaning
STOPPED	The run-time component is not running because it was shut down normally.
UNKNOWN	The status of run-time component cannot be determined.

### Related Commands

[cc get monitoring](#)

## Configuring the Platform Manager OSGi Profile Component

See the following sections:

- ["Configuration Types that OSGI-SPM-ENGINE Supports" on page 500](#)
- ["Run-time Monitoring Statuses for OSGI-SPM-ENGINE" on page 501](#)
- ["Run-time Monitoring States for OSGI-SPM-ENGINE" on page 501](#)

### Configuration Types that OSGI-SPM-ENGINE Supports

The OSGI-SPM-ENGINE run-time component supports creating instances of the configuration types listed in the following table.

Configuration Type	Use to configure...
COMMON-SYSPROPS	Monitoring service parameters, for example, the products' polling interval for run-time status and date.
SIN-INTERNAL-GROUPS	Internal user groups stored in the common/conf/groups.txt file.
SIN-INTERNAL-ROLES	User roles stored in the common/conf/roles.txt file.
SIN-INTERNAL-USERS	Internal users stored in the common/conf/users.txt file.
SPM-NODEID	Internal unique identifier for a Platform Manager.

**Note:** Command Central automatically manages unique identifiers. You can customize the identifiers. However, you *must* ensure that each

Configuration Type	Use to configure...
	<p>identifier is unique within the landscapes that Command Central manages.</p> <p>To view a list of identifiers already registered with Command Central, use the <a href="#">cc list landscape nodes</a> command.</p>

### Related Commands

[cc create configuration data](#)

[cc get configuration instances](#)

[cc list configuration instances](#)

[cc get configuration types](#)

[cc list configuration types](#)

## Run-time Monitoring Statuses for OSGI-SPM-ENGINE

The following table lists the run-time statuses that the OSGI-SPM-ENGINE run-time component can return in response to the `cc get monitoring runtimestatus` and `cc get monitoring state` commands, along with the meaning of each run-time status.

Run-time Status	Meaning
ONLINE	The run-time component is running.
STOPPED	The run-time component is not running because it was shut down normally.
UNKNOWN	The status of run-time component cannot be determined.

### Related Commands

[cc get monitoring](#)

## Run-time Monitoring States for OSGI-SPM-ENGINE

In response to the `cc get monitoring runtimestate` and `cc get monitoring state` commands, OSGI-SPM-ENGINE run-time component provides information about the key performance indicators (KPIs) in the following table.

KPI	Description
Computer Memory (MB)	<p>Use this KPI to monitor the memory usage of the computer where Platform Manager is running so that you can take corrective actions if memory use approaches a critical value.</p> <p>The KPI uses the following marginal, critical, and maximum values:</p> <ul style="list-style-type: none"> <li>■ Marginal is 80% of the maximum amount of physical memory.</li> <li>■ Critical is 95% of the maximum amount of physical memory.</li> <li>■ Maximum is the total amount of physical memory.</li> </ul>
Computer disk space (MB)	<p>Use this KPI to monitor the available disk space of the computer where Platform Manager is running so that you can take corrective actions if disk space usage approaches a critical value.</p> <p>The KPI uses the following marginal, critical, and maximum values:</p> <ul style="list-style-type: none"> <li>■ Marginal is 80% of the maximum amount of physical disk space.</li> <li>■ Critical is 95% of the maximum amount of physical disk space.</li> <li>■ Maximum the total amount of physical disk space.</li> </ul>
Computer CPU utilization	<p>Use this KPI to monitor the CPU usage of the computer where Platform Manager is running so that you can take corrective actions if CPU usage approaches a critical value.</p> <p>The KPI uses the following marginal, critical, and maximum values:</p> <ul style="list-style-type: none"> <li>■ Marginal is 80% of CPU utilization.</li> <li>■ Critical is 95% of CPU utilization.</li> <li>■ Maximum 100% of CPU utilization.</li> </ul>

### Related Commands

[cc get monitoring](#)

## Configuring the Tomcat OSGi Profile Components

This component runs in multiple profiles, for example:

- OSGI-CTP-TOMCAT-ENGINE for the Software AG Runtime OSGi profile
- OSGI-CCE-TOMCAT-ENGINE for the Command Central OSGi profile
- OSGI-SPM-TOMCAT-ENGINE for the Platform Manager OSGi profile

See the following sections:

- ["Lifecycle Actions for OSGI-\\*-TOMCAT-ENGINE" on page 503](#)
- ["Run-time Monitoring Statuses for OSGI-\\*-TOMCAT-ENGINE" on page 504](#)

### Lifecycle Actions for OSGI-\*-TOMCAT-ENGINE

The following table lists the run-time statuses that the OSGI-\*-TOMCAT-ENGINE run-time component can return in response to the `cc get monitoring runtimestatus` and `cc get monitoring state` commands, along with the meaning of each run-time status.

Action	Description
start	Starts the run-time component. When successful, the run-time status is set to ONLINE.  <b>Important</b> Do not use the <code>start</code> action for either the OSGI-CCE-TOMCAT-ENGINE or OSGI-SPM-TOMCAT-ENGINE run-time components.
restart	Stops, then restarts the run-time component. When successful, the run-time status is set to ONLINE.
stop	Stops the run-time component. When successful, the run-time status is set to STOPPED.  <b>Important</b> Do not use the <code>stop</code> action for either the OSGI-CCE-TOMCAT-ENGINE or OSGI-SPM-TOMCAT-ENGINE run-time components. Stopping the component ends remote communications with the Web user interface and the REST API.

#### Related Commands

[cc exec lifecycle](#)

---

## Run-time Monitoring Statuses for OSGI-\*-TOMCAT-ENGINE

The following table lists the run-time statuses that the OSGI-\*-TOMCAT-ENGINE run-time component can return in response to the `cc get monitoring runtimestatus` and `cc get monitoring state` commands, along with the meaning of each run-time status.

Run-time Status	Meaning
ONLINE	The OSGI-*-TOMCAT-ENGINE run-time component is running.
STOPPED	The OSGI-*-TOMCAT-ENGINE run-time component is not running because it was shut down normally.
UNKNOWN	The status of the OSGI-*-TOMCAT-ENGINE run-time component cannot be determined.

### Related Commands

[cc get monitoring](#)

---

# 22 Administering Task Engine

---

■ Overview .....	506
■ Commands that Task Engine Supports .....	506
■ Configuration Types that Task Engine Supports .....	508
■ Run-Time Monitoring Statuses for Task Engine .....	508

## Overview

You use the Command Central web user interface and command line interface to manage Task Engine. The Task Engine run-time component is a layered product of My webMethods Server.

You can use Command Central to manage the following operations for Task Engine:

- Monitor run-time statuses for a Task Engine instance.  
For information about monitoring a product instance, see ["Monitoring Instances" on page 221](#). For information about the Task Engine run-time statuses, see ["Run-Time Monitoring Statuses for Task Engine " on page 508](#).
- Configure an endpoint connection to webMethods Process Engine on Integration Server for a Task Engine instance. Task Engine is installed with pre-configured default values for the available fields of the Endpoints configuration. You can edit the fields to specify values other than the default ones.  
For more information about the fields and values to specify when configuring a Task Engine endpoint, see *webMethods Task Engine User's Guide*.  
For detailed steps about how to edit and test an endpoint configuration, see ["Configuring Instances" on page 242](#) and ["Testing a Configuration" on page 243](#).  
For information about the configuration commands that Task Engine supports, see ["Commands that Task Engine Supports" on page 506](#).

## Commands that Task Engine Supports

Task Engine supports the Platform Manager commands listed in the following table. The table lists where you can find information about each command.

Commands	For more information, see...
<code>cc get configuration data</code>	For general information about the command, see <a href="#">"cc get configuration data" on page 253</a> . For information about the configuration types that Task Engine supports, see <a href="#">"Configuration Types that Task Engine Supports" on page 508</a> .
<code>cc update configuration data</code>	For general information about the command, see <a href="#">"cc update configuration data" on page 256</a> .

Commands	For more information, see...
	<p>For information about the configuration types that Task Engine supports, see <a href="#">"Configuration Types that Task Engine Supports"</a> on page 508.</p>
<code>cc get configuration instances</code>	<p>For general information about the command, see <a href="#">"cc get configuration instances"</a> on page 260.</p> <p>For information about the configuration types that Task Engine supports, see <a href="#">"Configuration Types that Task Engine Supports"</a> on page 508.</p>
<code>cc list configuration instances</code>	<p>For general information about the command, see <a href="#">"cc list configuration instances"</a> on page 263.</p> <p>For information about the configuration types that Task Engine supports, see <a href="#">"Configuration Types that Task Engine Supports"</a> on page 508.</p>
<code>cc get configuration types</code>	<p>For general information about the command, see <a href="#">"cc get configuration types"</a> on page 265.</p> <p>For information about the configuration types that Task Engine supports, see <a href="#">"Configuration Types that Task Engine Supports"</a> on page 508.</p>
<code>cc list configuration types</code>	<p>For general information about the command, see <a href="#">"cc list configuration types"</a> on page 267.</p> <p>For information about the configuration types that Task Engine supports, see <a href="#">"Configuration Types that Task Engine Supports"</a> on page 508.</p>
<code>cc exec configuration validation update</code>	<p>For general information about the command, see <a href="#">"cc exec configuration validation update"</a> on page 274.</p> <p>For information about the configuration types that Task Engine supports, see</p>

Commands	For more information, see...
	<a href="#">"Configuration Types that Task Engine Supports"</a> on page 508.
<code>cc get monitoring</code>	<p>For general information about the command, see <a href="#">"cc get monitoring"</a> on page 227.</p> <p>For Task Engine-specific information about the command, see <a href="#">"Run-Time Monitoring Statuses for Task Engine "</a> on page 508.</p>

## Configuration Types that Task Engine Supports

The Task Engine run-time component supports the following configuration types.

Configuration Type	Use to configure...
COMMON-COMPONENT-ENDPOINTS	<p>A connection between Task Engine on My webMethods Server to webMethods Process Engine on Integration Server.</p> <p>For general information about using Command Central to configure settings for a product, see <a href="#">"Configuring Instances"</a> on page 242 and <a href="#">"Configuration Commands"</a> on page 244.</p> <p>For information about the fields and values to specify when configuring a Task Engine endpoint, see <i>webMethods Task Engine User's Guide</i></p>

## Run-Time Monitoring Statuses for Task Engine

The following table lists the run-time statuses that the Task Engine run-time component can return in response to the `cc get monitoring runtimestatus` and `cc get monitoring state` commands, along with the meaning of each run-time status.

Run-time Status	Meaning
ONLINE	The Task Engine instance is running.

Run-time Status	Meaning
STOPPED	The Task Engine instance is stopped.

---



---

# 23 Administering Universal Messaging

---

■ About Administering Universal Messaging .....	512
■ Universal Messaging Configuration Types .....	512
■ Universal Messaging License Configuration .....	513
■ Universal Messaging Memory Configuration .....	513
■ Universal Messaging Ports Configuration .....	513
■ Universal Messaging Authentication Configuration .....	515
■ Universal Messaging Users Configuration .....	516
■ Universal Messaging Cluster Management .....	516
■ Universal Messaging Inventory .....	517
■ Universal Messaging Lifecycle Actions .....	517
■ Universal Messaging Run-time Monitoring Statuses .....	518
■ Universal Messaging KPIs .....	518
■ Universal Messaging and the Command Line Interface .....	519

## About Administering Universal Messaging

---

This section describes the details specific to Universal Messaging administration.

Command Central administers a Universal Messaging server instance by using one of the ports (interfaces) of the server. Command Central checks the ports (interfaces) of a Universal Messaging server instance in the following order and chooses the first port (interface) that connects with the server:

1. Interfaces that use HTTP protocol (nhp)
2. Interfaces that use socket protocol (nsp)
3. Interfaces that use HTTPS protocol (nhps)
4. Interfaces that use SSL protocol (nsps)

**Note:** Command Central will use a secured port (nhps and nsps) to connect with a Universal Messaging server instance only if the client-side certificates are not required for establishing the connection. Command Central does not use ports that use shared memory protocol (shm).

At any point, if there is a disconnection between Command Central and the Universal Messaging server, Command Central will identify another port using the same order to check the next available port for communicating with the Universal Messaging server.

To perform advanced configuration tasks, use Universal Messaging Enterprise Manager. You cannot access Enterprise Manager through Command Central, but you can create Enterprise Manager instances using Universal Messaging instance management commands. For information about Enterprise Manager, see the Universal Messaging documentation.

## Universal Messaging Configuration Types

---

You can use the various configuration types that Command Central provides to configure the following settings on a Universal Messaging server instance:

- Clustering
- License
- Memory
- Ports
- Users

## Universal Messaging License Configuration

---

For a Universal Messaging server, you can configure the license, view the details of the license that is configured, and retrieve the location of the license file. You cannot change the location of a Universal Messaging license file.

## Universal Messaging Memory Configuration

---

You can view and update the initial memory size, maximum memory size, and the extended property (MaxDirectMemorySize) of a Universal Messaging server instance. When you add or edit an extended property, make sure you prefix `-xx:` to the extended property name.

## Universal Messaging Ports Configuration

---

You can view, create, enable, disable, edit the Universal Messaging server ports (interfaces). The port on which you install the Universal Messaging server is the primary port (interface) of the server. You cannot delete a primary port (interface).

**Note:** Command Central does not use or report the Universal Messaging server ports that use shared memory protocol (shm).

## Port Configuration Attributes

When you add a new port (interface), configure the attributes of the port. Set the security attributes for the secured ports that use either the HTTPS protocol or the SSL protocol.

### Basic Port Connection Attributes

The table describes the basic connection attributes of a port.

Configure this...	To specify...
Port Type	<p>The port (interface) should use one of the following protocols:</p> <ul style="list-style-type: none"> <li>■ Socket protocol (nsp)</li> <li>■ HTTP protocol (nhp)</li> <li>■ HTTPS protocol (nhps)</li> <li>■ SSL protocol (nsps)</li> </ul>

<u>Configure this...</u>	<u>To specify...</u>
	You cannot change this attribute after you create the port.
Port Number	The number of the port. You cannot change this attribute after you create the port.
Bind Address	The IP address to which to bind this port, if your machine has multiple IP addresses and you want the port to use this specific address. You cannot change this attribute after you create the port.
Backlog	The maximum size of the IP socket queue. When the incoming socket request queue reaches this maximum value, the incoming connection requests are refused. The requests will be serviced only when the queue size is less than the maximum size.
Enabled	Whether the port is enabled or disabled.  <b>Note:</b> If Command Central fails to enable a port, check the Universal Messaging logs to find out the reason for failure.

## Port Security Attributes

The table describes the security attributes you can configure for a secure SSL enabled port.

<u>Configure this...</u>	<u>To specify...</u>
Client Authentication	Whether or not Universal Messaging requires client certificates for all requests. Select: <ul style="list-style-type: none"> <li>■ <b>None</b> if Universal Messaging does not require client certificates for all requests.</li> <li>■ <b>REQUIRE_CERTIFICATE</b> if you want Universal Messaging to require client certificates for all requests.</li> </ul>
SSL Enabled	Whether the port is SSL enabled or not. This attribute is always set to true for nhps and nsps port.

Configure this...	To specify...
Keystore Type	File type of the keystore file. Universal Messaging supports only the JKS file type.
Server Location of Keystore	Location of the keystore file.
Keystore Password	Password required to access the SSL certificate in the keystore file.
Keystore Key Password	Password required to access a specific private key in the keystore file.
Truststore Type	File type of the truststore file. Universal Messaging supports only the JKS file type.
Server Location of Truststore	Location of the truststore file.
Truststore Password	Password required to access the SSL certificate in the truststore file.

## Universal Messaging Authentication Configuration

Perform the following steps to enable basic authentication for Universal Messaging server instance users.

1. Enable basic authentication in the Universal Messaging server instance. For information about security and authentication in Universal Messaging, see the Universal Messaging concepts documentation: <http://um.terracotta.org/>.
2. Use the user management feature of Command Central to add users to the user.txt file of the Universal Messaging server instance.
3. Use Universal Messaging Enterprise Manager to set the required ACLs for the Universal Messaging server instance users.
4. Restart the Universal Messaging server instance, if required.
5. Change the default Authentication Mode from **None** to **Fixed User** and provide the user name and password for the configured user. For more information, see "[Setting Outbound Authentication](#)" on page 150.

## Universal Messaging Users Configuration

---

Information to authenticate the users of a Universal Messaging server instance is stored in the user repository (users.txt file) of the Universal Messaging server instance. You can add new users, list the existing users, change the password of an user, or delete a user from the user repository.

The default path to the user repository (users.txt file) of a Universal Messaging server instance is *Software AG\_directory\UniversalMessaging\server\umserver\_instance*, where *umserver\_instance* is the name of the Universal Messaging server instance. The path to the users.txt file is specified in the Server\_Common.conf file of the server instance irrespective of whether you are running the Universal Messaging server instance as a service or as an application. If you specify a relative path, the users.txt file will be created in a directory relative to the bin directory of the Universal Messaging server instance.

You can also use the command line interface commands or internaluserrepo.bat/sh script to configure users of a Universal Messaging server instance. For more information, see *Software AG Command Central and Software AG Platform Manager Command Reference*.

## Universal Messaging Cluster Management

---

### Before You Create a Universal Messaging Cluster

Before you create or update a Universal Messaging cluster:

- Make sure the server instances that you want to add to the cluster are running.
- Verify that the permissions on the server machines allow connections to the other servers in the cluster.
- Make sure the /naming/defaultContext channel exists only on one or none of the nodes that will form the cluster. The Universal Messaging server instance used as a JNDI provider uses the /naming/defaultContext channel to store JMS references and JNDI objects. If channels exists on multiple nodes, you cannot create the cluster.

### Cluster Configuration Tasks Supported

Cluster configuration tasks that you can perform:

- Create a cluster of two or more server instances
- Add one or more server instances to the existing cluster
- Remove one or more server instances from the existing cluster
- Create sites and assign server instances to the sites

- Assign a site as the prime site of a cluster
- Remove one or more server instances from a cluster site
- Remove sites from a cluster
- Delete a cluster

## Cluster Configuration Fields

Field	Specify...
<b>Cluster Name</b>	Unique cluster name.
<b>Server URL</b>	Server instances URL (for example, <code>nsp://127.0.0.1:9002</code> ) of each server node.  When you save the cluster details, the <b>Server Name</b> field is populated with the name of the server corresponding to the specified server URL.
<b>Cluster Site</b>	Name of the site (Optional) to which the server node belongs.
<b>Prime Site</b>	Name of the primary site (Optional), if you have configured sites in the cluster.

## Universal Messaging Inventory

---

When you view installations in an environment, Command Central displays the Universal Messaging server instances listed in the *UniversalMessaging\_installationDirectory* \UniversalMessaging\server directory of an installation. Command Central lists all the folders (except the templates) in the server directory.

## Universal Messaging Lifecycle Actions

---

You can perform the following lifecycle actions on a Universal Messaging server instance.

- **Start.** Start a server instance that has stopped.
- **Stop.** Stop a running server instance.
- **Restart.** Restart a running server instance.

## Universal Messaging Run-time Monitoring Statuses

The run-time status of a Universal Messaging server instance states if the server is online, failed, stopped, unresponsive (when none of the server interfaces are connected to the server), or unknown. Universal Messaging does not report the starting and stopping statuses.

## Universal Messaging KPIs

This section describes the key performance indicators (KPIs) of Universal Messaging. These KPIs enable you to monitor the health of the Universal Messaging servers:

KPI	Description
JVM Memory	<p>Indicates the utilization of JVM memory.</p> <p>The marginal, critical, and maximum values for this KPI depend on the maximum memory size of the JVM.</p> <ul style="list-style-type: none"> <li>■ Marginal is 80% of the maximum JVM memory.</li> <li>■ Critical is 95% of the maximum JVM memory.</li> <li>■ Maximum is 100% of the maximum JVM memory.</li> </ul>
Fanout Backlog	<p>Indicates the total number of events currently waiting to be processed by the fanout engine. If the fanout backlog is more than the critical value, there is a possibility that the subscribers receive the published events after some delay.</p> <p>The KPI uses the following marginal, critical, and maximum values:</p> <ul style="list-style-type: none"> <li>■ Marginal is 80% of the maximum value.</li> <li>■ Critical is 95% of the maximum value.</li> <li>■ Maximum is 100% of the peak value (high-water mark) of fanout backlog. Default is 100.</li> </ul>
Queued Tasks	<p>Indicates the total number of tasks in the read, write, and common read/write pools. If the number of read and write tasks queued is more than the critical value, it indicates that the Universal Messaging server is unable to match the speed of the publishers and subscribers.</p>

KPI	Description
	<p>The KPI uses the following marginal, critical, and maximum values:</p> <ul style="list-style-type: none"> <li>■ Marginal is 80% of the maximum value.</li> <li>■ Critical is 95% of the maximum value.</li> <li>■ Maximum is 100% of the peak value (high-water mark) of read and write tasks queued. Default is 100.</li> </ul>

## Universal Messaging and the Command Line Interface

### Commands that Universal Messaging Server Instance Supports

Universal Messaging supports the Platform Manager commands listed in the following table. The table lists where you can find general information about each command. Additionally, if there is Universal Messaging-specific information, the table lists where you can learn more about arguments and options that Universal Messaging server instance supports or details about the actions Universal Messaging takes when you execute an `exec` command.

**Important:** When a Universal Messaging server instance is running as a service, you might not be able to perform some administrative tasks such as check the status, perform lifecycle actions, and configure the memory settings of the server instance.

Commands	Additional Information
<code>cc create configuration data</code>	<p>For general information about the command, see <a href="#">"cc create configuration data" on page 247</a>.</p> <p>For Universal Messaging-specific information about using this command, see <a href="#">"Configuration Types that Universal Messaging Server Instance Supports" on page 523</a>.</p>
<code>cc delete configuration data</code>	<p>For general information about the command, see <a href="#">"cc delete configuration data" on page 251</a>.</p>
<code>cc get configuration data</code>	<p>For general information about the command, see <a href="#">"cc get configuration data" on page 253</a>.</p>

Commands	Additional Information
	<p>For Universal Messaging-specific information about using this command, see <a href="#">"Configuration Types that Universal Messaging Server Instance Supports"</a> on page 523.</p>
<pre>cc update configuration data</pre>	<p>For general information about the command, see <a href="#">"cc update configuration data"</a> on page 256.</p>
<pre>cc get configuration instances</pre>	<p>For general information about the command, see <a href="#">"cc get configuration instances"</a> on page 260.</p> <p>For Universal Messaging-specific information about using this command, see <a href="#">"Configuration Types that Universal Messaging Server Instance Supports"</a> on page 523.</p>
<pre>cc list configuration instances</pre>	<p>For general information about the command, see <a href="#">"cc list configuration instances"</a> on page 263.</p> <p>For Universal Messaging-specific information about using this command, see <a href="#">"Configuration Types that Universal Messaging Server Instance Supports"</a> on page 523.</p>
<pre>cc get configuration types</pre>	<p>For general information about the command, see <a href="#">"cc get configuration types"</a> on page 265.</p> <p>For Universal Messaging-specific information about using this command, see <a href="#">"Configuration Types that Universal Messaging Server Instance Supports"</a> on page 523.</p>
<pre>cc list configuration types</pre>	<p>For general information about the command, see <a href="#">"cc list configuration types"</a> on page 267.</p> <p>For Universal Messaging-specific information about using this command, see <a href="#">"Configuration Types that Universal Messaging Server Instance Supports"</a> on page 523.</p>

Commands	Additional Information
	<p><a href="#">Messaging Server Instance Supports</a> on page 523.</p>
<p>cc exec configuration validation create</p>	<p>For general information about the command, see "<a href="#">cc exec configuration validation create</a>" on page 269.</p> <p>For Universal Messaging-specific information about using this command, see "<a href="#">Configuration Types that Universal Messaging Server Instance Supports</a>" on page 523.</p>
<p>cc exec configuration validation delete</p>	<p>For general information about the command, see "<a href="#">cc exec configuration validation delete</a>" on page 272.</p>
<p>cc exec configuration validation update</p>	<p>For general information about the command, see "<a href="#">cc exec configuration validation update</a>" on page 274.</p> <p>For Universal Messaging-specific information about using this command, see "<a href="#">Configuration Types that Universal Messaging Server Instance Supports</a>" on page 523.</p>
<p>cc create instances</p>	<p>For general information about the command, see "<a href="#">cc create instances</a>" on page 277.</p> <p>For Universal Messaging-specific information about using this command, see "<a href="#">Configuration Types that Universal Messaging Server Instance Supports</a>" on page 523.</p> <div data-bbox="743 1524 1373 1667" style="background-color: #f0f0f0; padding: 5px;"> <p><b>Note:</b> You cannot change the name of the Universal Messaging instance after you create it.</p> </div>
<p>cc delete instances</p>	<p>For general information about the command, see "<a href="#">cc delete instances</a>" on page 279.</p> <div data-bbox="743 1772 1373 1877" style="background-color: #f0f0f0; padding: 5px;"> <p><b>Important:</b> You must stop the Universal Messaging instance before deleting the instance.</p> </div>

Commands	Additional Information
<code>cc list instances</code>	For general information about the command, see <a href="#">"cc list instances"</a> on page 281.
<code>cc list instances supportedproducts</code>	For general information about the command, see <a href="#">"cc list instances supported products"</a> on page 282.
<code>cc get inventory components</code>	For general information about the command, see <a href="#">"cc get inventory components"</a> on page 194.
<code>cc list inventory components</code>	For general information about the command, see <a href="#">"cc list inventory components"</a> on page 198.  For Universal Messaging-specific information about using this command, see <a href="#">" Universal Messaging Server Instance Inventory"</a> on page 524.
<code>cc exec lifecycle</code>	For general information about the command, see <a href="#">"cc exec lifecycle"</a> on page 236.  For Universal Messaging-specific information about using this command, see <a href="#">" Universal Messaging Server Instance Lifecycle Actions"</a> on page 533.
<code>cc get diagnostics logs</code>	For general information about the command, see <a href="#">"cc get diagnostics logs"</a> on page 384.
<code>cc get diagnostic logs export file</code>	For general information about the command, see <a href="#">"cc get diagnostic logs export file"</a> on page 388.
<code>cc list diagnostics logs</code>	For general information about the command, see <a href="#">"cc list diagnostics logs"</a> on page 389.
<code>cc get monitoring</code>	For general information about the command, see <a href="#">"cc get monitoring"</a> on page 227.  For Universal Messaging-specific information about using this command, see:

Commands	Additional Information
	<ul style="list-style-type: none"> <li>■ <a href="#">"Run-time Monitoring Statuses for a Universal Messaging Server Instance" on page 534</a></li> <li>■ <a href="#">"Run-time Monitoring States for a Universal Messaging Server Instance" on page 535</a></li> </ul>

## Configuration Types that Universal Messaging Server Instance Supports

The following table lists the configuration types that Universal Messaging server instance supports.

Configuration Type	Use to...
COMMON-CLUSTER	Configure an active/active Universal Messaging cluster.
COMMON-LICENSE	Configure the Universal Messaging-specific SagLic license file.
COMMON-LICLOC	View the location of a Universal Messaging server instance's license file.  You cannot change the location of the license file.
COMMON-LOCAL-USERS	Configure and manage users of a Universal Messaging server instance. COMMON-LOCALUSERS- <i>userId</i> supports configuring the user ID and password of each user. By default, the users have administrator privileges for the Universal Messaging server instance.
COMMON-MEMORY	Configure the size of the initial memory, maximum memory, and the maximum direct memory (extended property) of a Universal Messaging server instance.
COMMON-PORTS	Configure the Universal Messaging server interfaces.

Configuration Type	Use to...
	<p><b>Note:</b>You cannot change the protocol, bind address, port number, or alias of a port of an existing server interface.</p> <p><b>Note:</b>If you change the SSL certificates of a secured interface, you must restart the interface.</p>

### Related Commands

[cc get configuration data](#)

[cc create configuration data](#)

[cc update configuration data](#)

[cc delete configuration data](#)

[cc get configuration instances](#)

[cc list configuration instances](#)

[cc get configuration types](#)

[cc list configuration types](#)

## Universal Messaging Server Instance Inventory

The following table lists the information you can retrieve about the Universal Messaging server instances configured in the *Universal Messaging\_directory*\server directory in an installation. Universal Messaging returns all the folders under the server directory, except the templates.

Property	Value
Display name	Universal-Messaging- <i>ServerInstanceName</i>
Run-time component ID	Universal-Messaging- <i>ServerInstanceName</i>
Product ID	NUMRealmServer
Run-time component category	PROCESS

### Related Commands

[cc get inventory components](#)

[cc list inventory components](#)

## Universal Messaging Server Instance License Management

You can view the content and the location of the license file, and change the license file of a Universal Messaging server instance.

### Examples When Executing on Command Central

- To view the license details of a Universal Messaging server instance with "Universal-Messaging-umserver" component ID that runs in the installation with alias name "sag01":

```
cc get configuration data sag01 Universal-Messaging-umserver
COMMON-LICENSE-Universal-Messaging
```

- To view the license file location of a Universal Messaging server instance with "Universal-Messaging-umserver" component ID that runs in the installation with alias name "sag01":

```
cc get configuration data sag01 Universal-Messaging-umserver
COMMON-LICLOC-Universal-Messaging
```

- To change the license file of a Universal Messaging server instance with "Universal-Messaging-umserver" component ID that runs in the installation with alias name "sag01":

```
cc update configuration data sag01 Universal-Messaging-umserver
COMMON-LICENSE-Universal-Messaging
-i C:\license\
  Universal_Messaging_directory
  \license.xml
```

## Universal Messaging Server Instance Memory Configuration

Use the `cc get configuration data` and `cc update configuration data` commands to view and configure a Universal Messaging server instance.

### Examples When Executing on Command Central

- To retrieve the memory settings of umserver Universal Messaging server instance

```
cc get configuration data sag01 Universal-Messaging-umserver COMMON-MEMORY
```

- To update the memory settings of umserver Universal Messaging server instance as specified in the umserverMemory.xml file

```
cc update configuration data sag01 Universal-Messaging-umserver
COMMON-MEMORY --input c:\umserverMemory.xml
```

## Universal Messaging Server Instance Management

The following table lists the required parameters that you must include when managing the Universal Messaging instances using the Universal Messaging instance management commands:

Command	Parameter	Description
<code>cc create instances</code>	NumRealmServer, NumEnterpriseManager, and NumTemplateApplications	The product ID for the Universal Messaging instance.
	<code>instance.name=name</code>	Required. A name for the new Universal Messaging instance.
	<code>instance.ip=ip_address</code>	Optional. An IP address for the Universal Messaging server interface. If you do not specify a value, <code>0.0.0.0</code> is the default value.
	<code>instance.port=port</code>	Optional. A port number for the Universal Messaging server. If you do not specify a value, <code>9000</code> is the default port number.
	<code>instance.dataDir</code>	Optional. Absolute path to Universal Messaging server data directory. If you do not specify a path, the default directory location is used for creating the data directory. Location of the default directory is: <i>Universal Messaging_directory\server\umserver\data</i>

**Important:** Universal Messaging does not support `cc update instances`. You cannot rename a Universal Messaging instance.

## Examples When Executing on Command Central

- To check if Universal Messaging supports instance management operations through Command Central for a node with alias “messagingNode”:

```
cc list instances messagingNode supportedproducts
```

- To create an instance of Universal Messaging server called “umserver” on port number “9000” of the node with alias “messagingNode”:

```
cc create instances messagingNode NUMRealmServer instance.name=umserver
instance.ip=0.0.0.0 instance.port=9000
```

- To create an instance of Universal Messaging server called “umserver” on port number “9000” of the node with alias “messagingNode” and having a custom data directory path:

```
cc create instances messagingNode NUMRealmServer instance.name=umserver
instance.ip=0.0.0.0 instance.port=9000
instance.dataDir=Data_Directory_Absolute_Path
```

- To read the properties such as the server name, port number, and the interface IP address of “umserver” Universal Messaging server instance existing on the node with alias “messagingNode”:

```
cc list instances messagingNode Universal-Messaging-umserver
```

- To remove the “umserver” Universal Messaging server instance existing on the node with alias “messagingNode” node:

```
cc delete instances messagingNode Universal-Messaging-umserver
```

- To create an Enterprise Manager instance named “EM1”:

```
cc create instances messagingNode instance.name=em1
instance.ip=0.0.0.0 instance.port=9000
```

Use the Universal Messaging Instance Manager tool to delete the Enterprise Manager and Template Applications instances. You cannot use Command Central commands for deleting these instances as they are not listed in the product inventory.

**Note:** You can create a Template Applications instance using the same command used to create an Enterprise Manager instance.

[cc create instances](#)

[cc delete instances](#)

[cc list instances supported products](#)

## Related Commands

## Universal Messaging Server Instance User Management

Information to authenticate the users of a Universal Messaging server instance is stored in the user repository (users.txt file).

You can use the command line interface commands, web user interface, or `internaluserrepo.bat/sh` script to configure users of a Universal Messaging server instance.

The user management operations you can perform on a Universal Messaging server instance:

- List the users existing in the user repository
- Change the password of an existing user
- Add new users to the user repository
- Delete existing users from the user repository

The path of the user repository, `users.txt` file is specified in the `Server_Common.conf` file of the Universal Messaging server instance irrespective of whether you are running Universal Messaging as a service or as an application. The default path for the Universal Messaging user repository (`users.txt` file) is `Universal Messaging_directory\server\umserver_instance`. While creating the user repository, if you specify a relative path, the `users.txt` file will be created in a directory relative to the `bin` directory of the Universal Messaging server instance.

### Examples When Executing on Command Central

- To list the path of the user repository and the users of a Universal Messaging server instance:

```
cc get configuration instances sag01 Universal-Messaging-umserver
```

where `umserver` is the name of the Universal Messaging server instance, and `sag01` is the alias name of the installation where `umserver` is running.

- To retrieve information of a Universal Messaging server instance user:

```
cc get configuration instances sag01 Universal-Messaging-umserver
COMMON-LOCAL-USERS-user1
```

where `umserver` is the name of the Universal Messaging server instance, `sag01` is the alias name of the installation where `umserver` is running, and `user1` is the user ID of the user.

- To add a user to a Universal Messaging server instance:

```
cc create configuration data sag01 Universal-Messaging-umserver
COMMON-LOCAL-USERS COMMON-LOCAL-USERS --input c:\input\xmls\user2.xml
```

where `umserver` is the name of the Universal Messaging server instance, `sag01` is the alias name of the installation where `umserver` is running, and `user2.xml` is file that contains the user ID and the password of the new user.

Format of the `user2.xml` file:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<User id="user2">
<Password>test</Password>
</User>
```

- To update the password of a Universal Messaging server instance user:

```
cc update configuration data sag01 Universal-Messaging-umserver
COMMON-LOCAL-USERS COMMON-LOCAL-USERS-user2 --input c:\inputxmls\user2.xml
```

where `umserver` is the name of the Universal Messaging server instance, `sag01` is the alias name of the installation where `umserver` is running, and `user2.xml` is the file that contains the new password of the specified user.

- To delete a Universal Messaging server instance user:

```
cc delete configuration data sag01 Universal-Messaging-umserver
COMMON-LOCAL-USERS COMMON-LOCAL-USERS-user2
```

where `umserver` is the name of the Universal Messaging server instance, `sag01` is the alias name of the installation where `umserver` is running.

### Related Commands

[cc create configuration data](#)

[cc delete configuration data](#)

[cc update configuration data](#)

[cc get configuration instances](#)

### internaluserrepo Script

The `internaluserrepo.bat/sh` script located in the *Software AG\_directory*\common\bin directory creates or modifies the `users.txt` file, adds and deletes users in the file, and changes specified internal user passwords.

For more information about the `internaluserrepo.bat/sh` script, see "[internaluserrepo Script](#)" on page 171.

### Examples When Executing on Command Central

- To create the first user, `user1` for the Universal-Messaging-umserver instance:

```
internaluserrepo.bat -f c:\
Universal Messaging_directory
\server\umserver\users.txt
-c -p user1Password user1
```

- To add `user2` to the user repository of the Universal-Messaging-umserver instance:

```
internaluserrepo.bat -f c:\
Universal Messaging_directory
\server\umserver\users.txt
-p user2Password user2
```

- To change the password of `user1` existing on the user repository of the Universal-Messaging-umserver instance:

```
internaluserrepo.bat -f c:\
Universal Messaging_directory
\server\umserver\users.txt
-p user1NewPassword user1
```

- To delete user 2 from the user repository of the Universal-Messaging-umserver instance:

```
internaluserrepo.bat -f c:\
Universal Messaging_directory
\server\umserver\users.txt
-d user2
```

- To check if user1 exists in the user repository of the Universal-Messaging-umserver instance:

```
internaluserrepo.bat -f c:\
Universal Messaging_directory
\server\umserver\users.txt
-e user1
```

## Universal Messaging Cluster Management

### Before You Create or Update a Universal Messaging Server Instance Cluster

- Make sure the server instances that you want to add to the cluster are running.
- Verify that the permissions on the server machines allow connections to the other servers in the cluster.
- Make sure the /naming/defaultContext channel exists only on one or none of the nodes that will form the cluster. The Universal Messaging server instance used as a JNDI provider uses the /naming/defaultContext channel to store JMS references and JNDI objects. If channels exists on multiple nodes, you cannot create the cluster.

### Create a Universal Messaging Cluster

To create an active/active cluster of Universal Messaging server instances, input the cluster configuration details in an XML file to the `cc create configuration data` command.

Configure...	By specifying...
Cluster name	Cluster name that is unique to an installation.
Server instances	Name, URL, and port of each server node.
Sites (Optional)	Name of the site to which each server node belongs. <code>siteName</code> is a server level property.
Primary site (Optional)	Name of the primary site, if you have configured sites in the cluster. <code>primeSite</code> is a cluster level property that holds the name of the site, which is flagged as <code>isPrime</code> .

## Examples When Executing on Command Central

To create a new cluster with the following configurations specified in the `umSalesClusterConfig.xml` file.

- Cluster name: `umSales`
- Cluster sites: `site1` and `site2`
- Primary site: `site1`
- Server instances in `site1`: `um9000`, `um9001`
- Server instances in `site2`: `um9002`, `um9003`

```
cc create configuration data sag01 Universal-Messaging-um9001
COMMON-CLUSTER --input C:\inputxmls\umSalesClusterConfig.xml
```

where `sag01` is the alias name of the installation where `Universal-Messaging-um9001` server instance is running. The cluster configurations are specified in the `umSalesClusterConfig.xml` file as shown:

```
<?xml version="1.0" encoding="UTF-8"?>
<ClusterSettings>
  <Name>umSales</Name>
  <Servers>
    <Server name="um9000">
      <URL>nsp://127.0.0.1:9000</URL>
      <ExtendedProperties>
        <Property name="siteName">site1</Property>
      </ExtendedProperties>
    </Server>
    <Server name="um9001">
      <URL>nsp://127.0.0.1:9001</URL>
      <ExtendedProperties>
        <Property name="siteName">site1</Property>
      </ExtendedProperties>
    </Server>
    <Server name="um9002">
      <URL>nsp://127.0.0.1:9002</URL>
      <ExtendedProperties>
        <Property name="siteName">site2</Property>
      </ExtendedProperties>
    </Server>
    <Server name="um9003">
      <URL>nsp://127.0.0.1:9003</URL>
      <ExtendedProperties>
        <Property name="siteName">site2</Property>
      </ExtendedProperties>
    </Server>
  </Servers>
  <ExtendedProperties>
    <Property name="primeSite">site1</Property>
  </ExtendedProperties>
</ClusterSettings>
```

## Get Universal Messaging Cluster Details

To retrieve the following details of the Universal Messaging cluster in an XML file, specify one of the server instances of the cluster in the `cc get configuration data` command.

- Name of the cluster
- Name, URL, and port of each Universal Messaging server instance in the cluster
- Site information, if sites are configured

### Example When Executing on Command Central

To view the details of the cluster configuration of the `um9001` Universal Messaging server instance:

```
cc get configuration data sag01 Universal-Messaging-um9001 COMMON-CLUSTER-um9001
```

where `sag01` is the alias name of the installation where `Universal-Messaging-um9001` server instance is running.

## Update Universal Messaging Cluster

The XML file used for configuring a cluster must contain all the specifications for the cluster. When you update a cluster, you only edit the parameters that specify the change; other specifications in the cluster configuration file should not be changed. You can make these Universal Messaging cluster configurations changes:

To...	Edit the cluster configuration XML file to...
Add one or more server instances to the existing cluster	Include the name, URL, and port of the server instances that you want to add to the cluster.
Remove one or more server instances from the existing cluster	Remove the specifications of the server instances that you want to remove from the existing cluster.
Create sites and assign server instances to sites	Set the <code>siteName</code> extended property of the corresponding server instances.
Assign a site as the prime site of a cluster	Assign the name of the prime site to the <code>primeSite</code> cluster level property.
Remove one or more server instances from a cluster site	Remove the <code>siteName</code> extended property of the corresponding server instances.

To...	Edit the cluster configuration XML file to...
Remove sites from a cluster	Remove the site definitions of all the server instances in the cluster.

### Usage Notes

A Universal Messaging server instance can be part of only one cluster.

If you remove all the server instances from a site, the site will be deleted. Server instance deletion is not allowed if the deletion operation leaves less than two server instances in the cluster

### Examples When Executing on Command Central

To update the configuration of the cluster that contains the um9001 server instance:

```
cc update configuration data sag01 Universal-Messaging-um9001
COMMON-CLUSTER-um9001 --input C:\inputxmls\umSalesClusterConfig.xml
```

where `sag01` is the alias name of the installation where `Universal-Messaging-um9001` server instance is running.

### Delete Universal Messaging Cluster

You delete the cluster by specifying one of the server instances of the cluster by using the `cc delete configuration data` command.

### Examples When Executing on Command Central

To delete the cluster that contains the um9001 server instance:

```
cc delete configuration data sag01 Universal-Messaging-um9001
COMMON-CLUSTER-um9001
```

where `sag01` is the alias name of the installation where `Universal-Messaging-um9001` server instance is running.

## Universal Messaging Server Instance Lifecycle Actions

The following table lists the actions that Universal Messaging supports with the `cc exec lifecycle` command and the operation taken against a Universal Messaging server when an action is executed.

Action	Description
start	Starts the Universal Messaging server instance. When successful, the Universal Messaging server instance run-time status is set to ONLINE.

Action	Description
<code>stop</code>	Stops the Universal Messaging server instance. The Universal Messaging server run-time status is STOPPED.
<code>restart</code>	Stops, then restarts the Universal Messaging server instance. The Universal Messaging server run-time status is set to ONLINE.

### Related Commands

[cc exec lifecycle](#)

## Run-time Monitoring Statuses for a Universal Messaging Server Instance

The following table lists the run-time statuses that Universal Messaging server instance can return in response to the `cc get monitoring runtimestatus` command, along with the meaning of each run-time status.

Universal Messaging server instance does not return the STARTING and STOPPING statuses.

Run-time Status	Meaning
ONLINE	Universal Messaging server instance is running.
FAILED	Universal Messaging server instance is not running due to some failure. LOCK file exists.
STOPPED	Universal Messaging server instance is not running because it was shut down normally. LOCK file does not exist.
UNRESPONSIVE	Universal Messaging server instance does not respond to a ping operation. LOCK file exists and the Universal Messaging server instance is running.
UNKNOWN	The status of Universal Messaging server instance cannot be determined.

**Related Commands**[cc get monitoring](#)**Run-time Monitoring States for a Universal Messaging Server Instance**

In response to the `cc get monitoring rntimestate` and `cc get monitoring state` commands, Universal Messaging server instance provides information about the following key performance indicators (KPIs):

KPI	Description
JVM memory usage	<p>Indicates the utilization of JVM memory.</p> <p>The KPI uses the following marginal, critical, and maximum values:</p> <ul style="list-style-type: none"> <li>■ Marginal is 80% of the maximum JVM memory.</li> <li>■ Critical is 95% of the maximum JVM memory.</li> <li>■ Maximum is 100% of the maximum JVM memory.</li> </ul>
Fanout backlog	<p>Indicates the total number of events currently waiting to be processed by the fanout engine. If the fanout backlog is more than the critical value, there is a possibility that the subscribers receive the published events after some delay.</p> <p>The KPI uses the following marginal, critical, and maximum values:</p> <ul style="list-style-type: none"> <li>■ Marginal is 80% of the maximum value.</li> <li>■ Critical is 95% of the maximum value.</li> <li>■ Maximum is 100% of the peak value (high-water mark) of fanout backlog. Default is 100.</li> </ul>
Tasks queued for read and write	<p>Indicates the total number of tasks in the read, write, and common read/write pools. If the number of read and write tasks queued is more than the critical value, it indicates that the Universal Messaging server instance is unable to match the speed of the publishers and subscribers.</p> <p>The KPI uses the following marginal, critical, and maximum values:</p>

KPI	Description
	<ul style="list-style-type: none"><li data-bbox="597 323 1159 359">■ Marginal is 80% of the maximum value.</li><li data-bbox="597 380 1133 415">■ Critical is 95% of the maximum value.</li><li data-bbox="597 436 1357 499">■ Maximum is 100% of the peak value (high-water mark) of read and write tasks queued. Default is 100.</li></ul>

---

For more information about the Universal Messaging server instance KPIs, see information about monitoring Universal Messaging server instance in "[Universal Messaging Run-time Monitoring Statuses](#)" on page 518.

#### **Related Commands**

[cc get monitoring](#)

---

# 24 Administering WmTaskClient

---

■ Overview .....	538
■ Commands that WmTaskClient Supports .....	538
■ Configuration Types that WmTaskClient Supports .....	540
■ Run-Time Monitoring Statuses for WmTaskClient .....	541

## Overview

You use the Command Central web user interface and command line interface to manage WmTaskClient. WmTaskClient is a layered product of webMethods Integration Server.

You can use Command Central to manage the following operations for WmTaskClient:

- Start and stop a WmTaskClient instance.  
For detailed information about changing the status of a product instance, see ["Starting, Stopping, Pausing, Resuming, and Debugging Instances" on page 234](#).
- Monitor run-time statuses for a WmTaskClient instance.  
For information about monitoring a product instance, see ["Monitoring Instances" on page 221](#). For information about the WmTaskClient run-time statuses, see ["Run-Time Monitoring Statuses for WmTaskClient" on page 541](#).
- Configure an endpoint connection to My webMethods Server for a WmTaskClient instance. WmTaskClient is installed with pre-configured default values for the available fields of the Endpoints configuration. You can edit the fields to specify values other than the default ones.  
  
For more information about the fields and values to specify when configuring a WmTaskClient endpoint, see *webMethods Task Engine User's Guide*.  
  
For detailed steps about how to edit and test an endpoint configuration, see ["Configuring Instances" on page 242](#) and ["Testing a Configuration" on page 243](#).  
  
For information about the configuration commands that WmTaskClient supports, see ["Commands that WmTaskClient Supports" on page 538](#).

## Commands that WmTaskClient Supports

WmTaskClient supports the Platform Manager commands listed in the following table. The table lists where you can find information about each command.

Commands	For more information, see...
<code>cc get configuration data</code>	For general information about the command, see <a href="#">"cc get configuration data" on page 253</a> .  For information about the configuration types that WmTaskClient supports, see <a href="#">"Configuration Types that WmTaskClient Supports" on page 540</a> .

Commands	For more information, see...
<code>cc update configuration data</code>	<p>For general information about the command, see <a href="#">"cc update configuration data" on page 256</a>.</p> <p>For information about the configuration types that WmTaskClient supports, see <a href="#">"Configuration Types that WmTaskClient Supports" on page 540</a>.</p>
<code>cc get configuration instances</code>	<p>For general information about the command, see <a href="#">"cc get configuration instances" on page 260</a>.</p> <p>For information about the configuration types that WmTaskClient supports, see <a href="#">"Configuration Types that WmTaskClient Supports" on page 540</a>.</p>
<code>cc list configuration instances</code>	<p>For general information about the command, see <a href="#">"cc list configuration instances" on page 263</a>.</p> <p>For information about the configuration types that WmTaskClient supports, see <a href="#">"Configuration Types that WmTaskClient Supports" on page 540</a>.</p>
<code>cc get configuration types</code>	<p>For general information about the command, see <a href="#">"cc get configuration types" on page 265</a>.</p> <p>For information about the configuration types that WmTaskClient supports, see <a href="#">"Configuration Types that WmTaskClient Supports" on page 540</a>.</p>
<code>cc list configuration types</code>	<p>For general information about the command, see <a href="#">"cc list configuration types" on page 267</a>.</p> <p>For information about the configuration types that WmTaskClient supports, see <a href="#">"Configuration Types that WmTaskClient Supports" on page 540</a>.</p>

Commands	For more information, see...
cc exec configuration validation update	<p>For general information about the command, see <a href="#">"cc exec configuration validation update"</a> on page 274.</p> <p>For information about the configuration types that WmTaskClient supports, see <a href="#">"Configuration Types that WmTaskClient Supports"</a> on page 540.</p>
cc exec lifecycle	<p>For general information about the command, see <a href="#">"cc exec lifecycle"</a> on page 236.</p> <p>For WmTaskClient-specific information about the command, see <a href="#">"Run-Time Monitoring Statuses for WmTaskClient"</a> on page 541.</p>
cc get monitoring	<p>For general information about the command, see <a href="#">"cc get monitoring"</a> on page 227.</p> <p>For WmTaskClient-specific information about the command, see <a href="#">"Run-Time Monitoring Statuses for WmTaskClient"</a> on page 541.</p>

## Configuration Types that WmTaskClient Supports

The WmTaskClient run-time component supports the following configuration types.

Configuration Type	Use to configure...
COMMON-COMPONENT-ENDPOINTS	<p>A connection between WmTaskClient and My webMethods Server.</p> <p>For general information about using Command Central to configure settings for a product, see <a href="#">"Configuring Instances"</a> on page 242 and <a href="#">"Configuration Commands"</a> on page 244.</p> <p>For information about the fields and values to specify when configuring a WmTaskClient</p>

---

Configuration Type	Use to configure...
	endpoint, see <i>webMethods Task Engine User's Guide</i>

---

## Run-Time Monitoring Statuses for WmTaskClient

---

The following table lists the run-time statuses that the WmTaskClient run-time component can return in response to the `cc get monitoring runtimestatus` and `cc get monitoring state` commands, along with the meaning of each run-time status.

Run-time Status	Meaning
ONLINE	The WmTaskClient instance is running.
STOPPED	The WmTaskClient instance is stopped.

---



---

# 25 Administering Terracotta

---

■ Commands that Terracotta Supports .....	544
■ Configuration Types that Terracotta Supports .....	545
■ Lifecycle Actions for Terracotta .....	545
■ Run-time Monitoring Statuses for Terracotta .....	546
■ Run-time Monitoring States for Terracotta .....	546

## Commands that Terracotta Supports

Terracotta supports the Command Central commands listed in the following table. The table lists where you can find general information about each command. Additionally, if there is Terracotta-specific information, the table lists where you can learn more about arguments and options that Terracotta supports or details about the actions Terracotta takes when you execute an `exec` command.

Commands	Additional Information
<code>cc get configuration data</code>	For general information about the command, see <a href="#">cc get configuration data</a> .  For Terracotta-specific information about configuration types, see " <a href="#">Configuration Types that Terracotta Supports</a> " on page 545.
<code>cc get configuration instances</code>	For general information about the command, see <a href="#">cc get configuration instances</a> .
<code>cc list configuration instances</code>	For general information about the command, see <a href="#">cc list configuration instances</a> .
<code>cc get configuration types</code>	For general information about the command, see <a href="#">cc get configuration types</a> .
<code>cc list configuration types</code>	For general information about the command, see <a href="#">cc list configuration types</a> .
<code>cc get diagnostic logs</code>	For general information about the command, see <a href="#">cc get diagnostics logs</a> .
<code>cc get monitoring</code>	For general information about the command, see <a href="#">cc get monitoring</a> .
<code>cc get inventory components</code>	For general information about the command, see " <a href="#">Inventory Commands</a> " on page 192.
<code>cc list inventory components</code>	For general information about the command, see " <a href="#">cc list inventory components</a> " on page 198.

Commands	Additional Information
<code>cc exec lifecycle</code>	<p>For general information about the command, see <a href="#">Lifecycle Commands</a>.</p> <p>For Terracotta-specific information about using this command, see "<a href="#">Configuration Types that Terracotta Supports</a>" on page 545.</p>

## Configuration Types that Terracotta Supports

The Terracotta run-time component supports creating instances of the configuration types listed in the following table.

Configuration Type	Use to...
COMMON-CLUSTER	View the configuration for a cluster.
COMMON-MEMORY	Configure the JVM memory settings.
COMMON-PORTS	View the custom port configuration for a Terracotta Server Array instance.
TC-CONFIG	Configure the tc-config.xml file.
TC-SERVER-SERVER	Assign a server name to a Terracotta Server Array instance that matches the server name in the tc-config.xml file.

## Lifecycle Actions for Terracotta

The following table lists the actions that Terracotta supports with the `cc exec lifecycle` command and the operation taken against Terracotta when an action is executed.

**Important:** You must execute all lifecycle operations in the correct order to ensure safe startup, shutdown, and restart for Terracotta Server Array cluster instances. For more information about the correct order of lifecycle operations, see BigMemory Max product documentation for 4.2 at [www.terracotta.org/documentation](http://www.terracotta.org/documentation).

Action	Description
Start	Start a server instance that has stopped.
Stop	Stop a running server instance.
Restart	Restart a running server instance.

## Run-time Monitoring Statuses for Terracotta

The following table lists the run-time statuses that the Terracotta run-time component can return in response to the `cc get monitoring runtimestatus` and `cc get monitoring state` commands, along with the meaning of each run-time status.

Run-time Status	Meaning
STOPPED	The server is not running.
ONLINE_MASTER	The server is running and is the master in its stripe.
ONLINE_SLAVE	The server is running and is a slave (mirror) in its stripe.
FAILED	The server was running, but crashed. A possible reason for this failure is that the server did not find the correct license.
UNRESPONSIVE	The server is running, but is not responding on the specified ports. A possible reason for this failure is that the server did not find the correct license.

## Run-time Monitoring States for Terracotta

In response to the `cc get monitoring runtimestate` and `cc get monitoring state` commands, the Terracotta run-time component provides information about the following key performance indicators (KPIs):

KPI	Description
Live Object Count	The number of live objects in the cache.

---

KPI	Description
Offheap	The off-heap storage capacity for the cache.
Write Operations Rate	The rate of write operations for the cache.

---

The KPIs apply to the whole Terracotta Server Array cluster.



---

# A Command Central Task Quick Reference

---

■ Working with Authentication between Command Central and Managed Products .....	550
■ Working with Environments .....	550
■ Working with Installations .....	552
■ Working with Templates .....	555
■ Working with Instances .....	555
■ Working with Logs .....	559
■ Working with Product Comparisons .....	560
■ Working with Product Inventory .....	561
■ Working with Repositories .....	562
■ Working with Configuration Types .....	564
■ Working with Ports .....	565
■ Working with KPIs .....	566
■ Working with Security Credentials .....	566
■ Working with Licenses .....	567

The following sections list common Command Central tasks and how to perform them using the Command Central web user interface and the Command Central command line interface.

For more information about the interface navigation and commands listed in this quick reference, including usage notes, arguments, and options, see the appropriate documentation for the interface you are using.

## Working with Authentication between Command Central and Managed Products

This section provides a quick reference to the Command Central tasks that pertain to authentication between Command Central and the products it manages.

### Changing the Fixed User Password

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Authentication Mode (select an <b>Environment</b> , click the <b>Instances</b> tab, select an <b>Instance</b> , click the <b>Overview</b> tab, click the  <b>Authentication Edit</b> button)
Command line interface	<code>cc add security credentials</code>

## Working with Environments

This section provides a quick reference to the Command Central tasks that pertain to managing environments.

### Searching for an Environment

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Environments pane (type search criteria in the <b>Search Environments</b> box)
Command line interface	<code>cc list landscape environments</code>

## Viewing Details about an Environment

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Environments pane (select an environment)
Command line interface	<code>cc get landscape environments</code> <code>cc list landscape environments</code>

## Adding an Environment

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Environments pane (click the  <b>Add Environment</b> button)
Command line interface	<code>cc create landscape environments</code>

## Editing Environment Details

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Environments pane (select an environment, click the  <b>Options</b> button in the Environments pane, select <b>Edit Environment</b> )
Command line interface	<code>cc update landscape environments</code>

## Deleting an Environment

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Environments pane (select an environment, click the  <b>Delete Environment</b> button)

If you are using this interface...	Use these navigation steps or commands...
Command line interface	<code>cc delete landscape environments</code>

## Working with Installations

This section provides a quick reference to the Command Central tasks that pertain to managing installations.

### Searching for an Installation

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Installations tab (select an <b>Environment</b> , click the <b>Installations</b> tab, type search criteria in the <b>Search Installations</b> box)
Command line interface	<code>cc list landscape nodes</code>

### Viewing Details about an Installation

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Installations tab (select an <b>Environment</b> , click the <b>Installations</b> tab, select an <b>Installation</b> )
Command line interface	<code>cc get landscape nodes</code> <code>cc list landscape nodes</code>

### Viewing Products Installed in an Installation

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Installation Products tab (select an <b>Environment</b> , click the <b>Installations</b> tab, select an <b>Installation</b> , click the <b>Products</b> tab)

If you are using this interface...	Use these navigation steps or commands...
Command line interface	<pre>cc get inventory products</pre> <pre>cc list inventory products</pre>

### Viewing Fixes Installed in an Installation

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Installation Fixes tab (select an <b>Environment</b> , click the <b>Installations</b> tab, select an <b>Installation</b> , click the <b>Fixes</b> tab)
Command line interface	<pre>cc list inventory fixes</pre>

### Adding an Installation

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Installations tab (select an <b>Environment</b> , click the <b>Installations</b> tab, click the  <b>Add Installation</b> button)
Command line interface	<pre>cc create landscape nodes</pre>

### Updating the Properties of an Installation

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Installation Overview tab (select an <b>Environment</b> , click the <b>Installations</b> tab, select an <b>Installation</b> , click the <b>Overview</b> tab)
Command line interface	<pre>cc update landscape nodes</pre>

### Monitoring the Status of Product Instances in an Installation

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Installation Overview tab (select an <b>Environment</b> , click the <b>Installations</b> tab, select an <b>Installation</b> , click the <b>Overview</b> tab)
Command line interface	<code>cc get monitoring</code>

### Linking an Installation to Multiple Environments

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Installations tab (select an <b>Environment</b> , click the <b>Installations</b> tab, drag an installation to the desired environment)
Command line interface	<code>cc add landscape environments nodes</code>

### Creating a Unique ID for an Existing Installation

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Not supported
Command line interface	<code>cc exec landscape nodes generateNodeId</code>

### Removing an Installation from an Environment

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Installations tab (select an <b>Environment</b> , click the <b>Installations</b> tab, select the installation to remove, click the  <b>Remove Installations</b> button)
Command line interface	<code>cc remove landscape environments nodes</code>

## Working with Templates

This section provides a quick reference to the Command Central tasks that pertain to template-based provisioning of products.

### Creating a Template from an Existing Installation

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Installations tab (select an <b>Environment</b> , click the <b>Installations</b> tab, select the installation to provision, click  and select <b>Save as template</b> .)
Command line interface	<code>cc create templates</code>

### Applying a Template to a New Node

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Installations tab (select an <b>Environment</b> , click the <b>Installations</b> tab, select the installation to provision, click  and select <b>Apply template</b> .)
Command line interface	<code>cc exec templates apply</code>

## Working with Instances

This section provides a quick reference to the Command Central tasks that pertain to managing instances.

### Viewing a List of Instances in an Environment

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Instances tab (select an <b>Environment</b> , click the <b>Instances</b> tab)

If you are using this interface...	Use these navigation steps or commands...
------------------------------------	-------------------------------------------

Command line interface	<code>cc list inventory components</code>
------------------------	-------------------------------------------

### Viewing Details about an Instance

If you are using this interface...	Use these navigation steps or commands...
------------------------------------	-------------------------------------------

Web user interface	Instances tab (select an <b>Environment</b> , click the <b>Instances</b> tab, select an <b>Instance</b> )
--------------------	-----------------------------------------------------------------------------------------------------------

Command line interface	<code>cc get monitoring</code> <code>cc get inventory components</code>
------------------------	----------------------------------------------------------------------------

### Changing the Display Name of an Instance

If you are using this interface...	Use these navigation steps or commands...
------------------------------------	-------------------------------------------

Web user interface	Instance Overview tab (select an <b>Environment</b> , click the <b>Instances</b> tab, select an <b>Instance</b> , click the <b>Overview</b> tab, update the <b>Display Name</b> field, press Enter)
--------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Command line interface	<code>cc update inventory components</code>
------------------------	---------------------------------------------

### Changing the Icon Representing an Instance

If you are using this interface...	Use these navigation steps or commands...
------------------------------------	-------------------------------------------

Web user interface	Instance Overview tab (select an <b>Environment</b> , click the <b>Instances</b> tab, select an <b>Instance</b> , click the <b>Overview</b> tab, click the arrow next to the display name icon in the Details section)
--------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Command line interface	<code>cc list resources icons</code> <code>cc update inventory components</code>
------------------------	-------------------------------------------------------------------------------------

## Viewing or Clearing Instance Alerts

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Instance Overview tab (select an <b>Environment</b> , click the <b>Instances</b> tab, select an <b>Instance</b> , click the <b>Overview</b> tab, point to the flag in the <b>Alerts</b> area to view alert details, click the number in the <b>Alerts</b> area to clear the alerts)
Command line interface	<code>cc list monitoring alerts</code> <code>cc delete monitoring alerts</code>

## Starting an Instance

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Instances tab (select an <b>Environment</b> , click the <b>Instances</b> tab, click the <b>Status</b> button next to the instance, select <b>Start</b> )
Command line interface	<code>cc exec lifecycle</code>

## Stopping an Instance

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Instances tab (select an <b>Environment</b> , click the <b>Instances</b> tab, click the <b>Status</b> button next to the instance, select <b>Stop</b> )
Command line interface	<code>cc exec lifecycle</code>

## Restarting an Instance

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Instances tab (select an <b>Environment</b> , click the <b>Instances</b> tab, click the <b>Status</b> button next to the instance, select <b>Restart</b> )
Command line interface	<code>cc exec lifecycle</code>

## Pausing an Instance

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Instances tab (select an <b>Environment</b> , click the <b>Instances</b> tab, click the <b>Status</b> button next to the instance, select <b>Pause</b> )
Command line interface	<code>cc exec lifecycle</code>

## Resuming an Instance

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Instances tab (select an <b>Environment</b> , click the <b>Instances</b> tab, click the <b>Status</b> button next to the instance, select <b>Resume</b> )
Command line interface	<code>cc exec lifecycle</code>

## Debugging an Instance

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Instances tab (select an <b>Environment</b> , click the <b>Instances</b> tab, click the <b>Status</b> button next to the instance, select <b>Debug</b> )

If you are using this interface...	Use these navigation steps or commands...
Command line interface	<code>cc exec lifecycle</code>

## Working with Logs

This section provides a quick reference to the Command Central tasks that pertain to viewing and downloading product logs.

### Viewing Product Logs

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Select an environment in the Environments pane, select the <b>Instances</b> tab, click the name of a product instance, select the <b>Logs</b> tab
Command line interface	<code>cc list diagnostics logs</code>

### Viewing the Contents of a Log

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Select an environment in the Environments pane, select the <b>Instances</b> tab, click the name of a product instance, select the <b>Logs</b> tab, click the log alias for a log in the Alias column
Command line interface	<code>cc get diagnostics logs</code>

### Downloading Logs

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Select an environment in the Environments pane, select the <b>Instances</b> tab, click the name of a product instance, select the <b>Logs</b> tab, select the logs you want

If you are using this interface...	Use these navigation steps or commands...
	to download, click  , click <b>Download selected logs</b>
Command line interface	<code>cc get diagnostic logs export file</code>

## Working with Product Comparisons

This section provides a quick reference to the Command Central tasks that pertain to comparing products in an installation.

### Comparing Product Versions

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Compare Products page (select an <b>Environment</b> , click the <b>Installations</b> tab, select two or more <b>Installations</b> , click the  <b>Options</b> button, select <b>Compare Products</b> )
Command line interface	<code>cc get inventory products compare</code>

### Comparing Fix Levels

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Compare Fixes page (select an <b>Environment</b> , click the <b>Installations</b> tab, select two or more <b>Installations</b> , click the  <b>Options</b> button, select <b>Compare Fixes</b> )
Command line interface	<code>cc get inventory fixes compare</code>

## Comparing Configuration Settings

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Compare Configurations page (select an <b>Environment</b> , click the <b>Instances</b> tab, select two or more <b>Instances</b> , click the  <b>Options</b> button, select <b>Compare Configuration</b> )
Command line interface	<code>cc get configuration compare</code>

## Working with Product Inventory

This section provides a quick reference to the Command Central tasks that pertain to maintaining product inventory.

### Viewing Component Inventory

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Instances tab (select an <b>Environment</b> , click the <b>Instances</b> tab, click the arrow to the left of an instance name)
Command line interface	<code>cc get inventory components</code> <code>cc list inventory components</code>

### Viewing Product Inventory

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Installations tab (select an <b>Environment</b> , click the <b>Installations</b> tab, click the arrow to the left of an installation name)
Command line interface	<code>cc get inventory products</code> <code>cc list inventory products</code>

## Viewing Fix Inventory

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Installation Fixes tab (select an <b>Environment</b> , click the <b>Installations</b> tab, select an <b>Installation</b> , click the <b>Fixes</b> tab)
Command line interface	<code>cc list inventory fixes</code>

## Working with Repositories

This section provides a quick reference to the Command Central tasks that pertain to managing repositories.

### Creating an Image Repository

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Go to <b>Views &gt; Repositories</b> , click  , select <b>Image</b> , specify values for the fields in the Create image repository dialog box
Command line interface	<code>cc create repository</code>

### Registering a Master Repository

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Go to <b>Views &gt; Repositories</b> , click  , select <b>Master</b> , specify values for the fields in the Register master repository dialog box
Command line interface	<code>cc create repository</code>

## Deleting a Repository

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Go to <b>Views &gt; Repositories</b> , select the repository to delete, click 
Command line interface	<pre>cc delete repository</pre> <pre>cc delete repository with node alias</pre> <pre>cc delete repositories</pre>

## Editing Image Repository Details

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Go to <b>Views &gt; Repositories</b> , click the name of the repository you want to edit, edit the fields that allow modifying
Command line interface	<pre>cc update repository</pre> <pre>cc update repository details</pre>

## Editing Master Repository Details

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Go to <b>Views &gt; Repositories</b> , click the name of the repository you want to edit, edit the fields that allow modifying
Command line interface	<pre>cc update repository</pre> <pre>cc update repository details</pre>

## Working with Configuration Types

This section provides a quick reference to the Command Central tasks that pertain to managing configuration types.

### Adding a Configuration Type

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Instance Configuration tab (select an <b>Environment</b> , click the <b>Instances</b> tab, select an <b>Instance</b> , click the <b>Configuration</b> tab, select a component from the list/tabs on the left, select a configuration type from the list at the top of the page, click  .
Command line interface	<pre>cc configuration types cc list configuration types cc list configuration instances cc get configuration data cc create configuration data cc add configuration data cc update configuration data cc delete configuration instance cc exec configuration validation</pre>

### Editing or Viewing Details for Configuration Types

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Instance Configuration tab (select an <b>Environment</b> , click the <b>Instances</b> tab, select an <b>Instance</b> , click the <b>Configuration</b> tab, select a component from the list/tabs on the left, select a configuration type from the list at the top of the page, click the name or ID of the configuration property you want to edit, click the <b>Edit</b> button.

If you are using this interface...	Use these navigation steps or commands...
Command line interface	<pre>cc get configuration common cc list configuration types cc get configuration types cc list configuration instances cc get configuration data cc create configuration data cc add configuration data cc update configuration data cc delete configuration instance</pre>

## Deleting a Configuration Type

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Instance Configuration tab (select an <b>Environment</b> , click the <b>Instances</b> tab, select an <b>Instance</b> , click the <b>Configuration</b> tab, select a component from the list/tabs on the left, select a configuration type from the list at the top of the page, select the row of the configuration you want to delete, click  .
Command line interface	<pre>cc exec configuration validation delete and then cc delete configuration data</pre>

## Working with Ports

This section provides a quick reference to the Command Central tasks that pertain to managing ports.

## Enabling or Disabling a Port

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Instance Configuration tab (select an <b>Environment</b> , click the <b>Instances</b> tab, select an <b>Instance</b> , click the <b>Configuration</b> tab, click the port number, click the <b>Edit</b> button)
Command line interface	<code>cc get configuration data</code> and then <code>cc update configuration data</code>

## Working with KPIs

This section provides a quick reference to the Command Central tasks that pertain to working with key performance indicators (KPIs).

### Viewing KPIs

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Instance Overview tab, Monitoring section (select an <b>Environment</b> , click the <b>Instances</b> tab, select an <b>Instance</b> , click the <b>Overview</b> tab)
Command line interface	<code>cc get monitoring</code>

## Working with Security Credentials

This section provides a quick reference to the Command Central tasks that pertain to managing security credentials.

## Retrieving Security Credentials

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Authentication Mode (select an <b>Environment</b> , click the <b>Instances</b> tab, select an <b>Instance</b> , click the <b>Overview</b> tab, click the  <b>Authentication Edit</b> button)
Command line interface	<code>cc get security credentials</code>

## Adding Security Credentials

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Authentication Mode (select an <b>Environment</b> , click the <b>Instances</b> tab, select an <b>Instance</b> , click the <b>Overview</b> tab, click the  <b>Authentication Edit</b> button)
Command line interface	<code>cc add security credentials</code>

## Deleting Security Credentials

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Not supported
Command line interface	<code>cc delete security credentials</code>

## Working with Licenses

This section provides a quick reference to the Command Central tasks that pertain to managing product licenses.

### Viewing Details about a License

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Licenses (select an <b>Environment</b> , click the <b>Instances</b> tab, select an <b>Instance</b> , click the <b>Configuration</b> tab, select <b>Licenses</b> from the list at the top of the page, select a license type)
Command line interface	<pre>cc get configuration common cc get configuration types cc list configuration types cc get configuration data</pre>

### Selecting a New License or Changing the Server License Location

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Licenses (select an <b>Environment</b> , click the <b>Instances</b> tab, select an <b>Instance</b> , click the <b>Configuration</b> tab, select <b>Licenses</b> from the list at the top of the page, select a license type, click <b>Edit</b> )
Command line interface	<pre>cc get data cc exec configuration validation update cc update configuration data</pre>

### Creating a License Report Snapshot

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Go to <b>Views &gt; Licenses</b> , click 
Command line interface	<pre>cc create license-tools reports snapshot</pre>

## Viewing Details About License Reports

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Go to <b>Views &gt; Licenses</b> , use the report ID to search for a specific report
Command line interface	<pre>cc license-tools reports snapshot cc get license- tools reports snapshot reportid</pre>

## Downloading License Reports

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Go to <b>Views &gt; Licenses</b> , select the license report you want to download, click  , select the appropriate format for download
Command line interface	<pre>cc get license- tools reports snapshot output PDF cc get license- tools reports snapshot output XML</pre>

## Deleting License Reports

If you are using this interface...	Use these navigation steps or commands...
Web user interface	Go to <b>Views &gt; Licenses</b> , select the license report you want to delete, click 
Command line interface	<pre>cc delete license-tools reports snapshot cc delete license-tools reports snapshot reportid</pre>



# B Supported Functions and Configurations for Different Product Versions

---

■ Broker .....	572
■ Business Rules .....	573
■ CentraSite .....	574
■ Cloud Factory .....	574
■ Event-Driven Architecture/NERV .....	575
■ Infrastructure Data Collector .....	576
■ Integration Server .....	577
■ Monitor .....	579
■ My webMethods Server .....	580
■ OneData .....	582
■ Optimize User Interface .....	583
■ Platform Manager .....	584
■ Task Engine .....	585
■ Terracotta .....	586
■ Universal Messaging .....	587

The tables in the following sections list the Command Central functions and types of configuration supported for different product versions. “Yes” in a version column indicates that this core function or configuration type is supported beginning with the column version to the latest Command Central version. The Configuration Types table lists *only* the configuration types that Command Central supports for a product.

## Broker

### Core Functions

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					
Installation					
Instances					
Inventory	Yes				
Lifecycle	Yes				
Log access					
Monitoring	Yes				
Snapshot templates		Yes			
SSO	Yes				
Status	Yes				

### Configuration Types

Configuration Type	9.0	9.5	9.6	9.7	9.8
Licenses	Yes				

Configuration Type	9.0	9.5	9.6	9.7	9.8
Ports	Yes				

## Business Rules

### Core Functions

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					Yes
Installation					Yes
Instances					
Inventory			Yes		
Lifecycle			Yes		
Log access					
Monitoring					
Snapshot templates				Yes	
SSO					
Status			Yes		

### Configuration Types

Configuration Type	9.0	9.5	9.6	9.7	9.8
Licenses				Yes	

## CentraSite

### Core Functions

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					
Installation					Yes
Instances					
Inventory	Yes				
Lifecycle	Yes				
Log access					
Monitoring					
Snapshot templates					
SSO					
Status	Yes				

### Configuration Types

None.

## Cloud Factory

### Core Functions

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					Yes

Function	9.0	9.5	9.6	9.7	9.8
Installation					Yes
Instances					
Inventory			Yes		
Lifecycle			Yes		
Log access					
Monitoring					
Snapshot templates			Yes		
SSO					
Status			Yes		

#### Configuration Type

Configuration Type	9.0	9.5	9.6	9.7	9.8
CFS Accounts			Yes		

## Event-Driven Architecture/NERV

#### Core Functions

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					
Installation					

Function	9.0	9.5	9.6	9.7	9.8
Instances					
Inventory		Yes			
Lifecycle		Yes			
Log access			Yes		
Monitoring					
Snapshot templates					
SSO					
Status		Yes			

### Configuration Type

Configuration Type	9.0	9.5	9.6	9.7	9.8
System Properties			Yes		

## Infrastructure Data Collector

### Core Functions

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					
Installation					Yes
Instances					

Function	9.0	9.5	9.6	9.7	9.8
Inventory	Yes				
Lifecycle	Yes				
Log access			Yes		
Monitoring	Yes				
Snapshot templates					
SSO					
Status	Yes				

### Configuration Types

Configuration Type	9.0	9.5	9.6	9.7	9.8
Memory				Yes	
Ports	Yes				
Proxy				Yes	

## Integration Server

### Core Functions

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					Yes
Installation					Yes
Instances			Yes		

Function	9.0	9.5	9.6	9.7	9.8
Inventory	Yes				
Lifecycle	Yes				
Log access			Yes		
Monitoring	Yes				
Snapshot templates		Yes			
SSO	Yes				
Status	Yes				

### Configuration Types

Configuration Type	9.0	9.5	9.6	9.7	9.8
Cluster				Yes	
Database connection		Yes			
Database functional aliases		Yes			
JMS aliases		Yes			
JNDI aliases		Yes			
Keystores		Yes			
Licenses	Yes				
Loggers			Yes		
Memory				Yes	

Configuration Type	9.0	9.5	9.6	9.7	9.8
Ports	Yes				
Proxy				Yes	
Remote servers					Yes
Resources			Yes		
SMTP settings		Yes			
System properties		Yes			
Truststores		Yes			
Users				Yes	
Variables			Yes		
Web service endpoints				Yes	
webMethods messaging			Yes		

## Monitor

### Core Functions

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					Yes
Installation					Yes

Function	9.0	9.5	9.6	9.7	9.8
Instances					
Inventory				Yes	
Lifecycle				Yes	
Log access					
Monitoring					
Snapshot templates		Yes			
SSO					
Status				Yes	

### Configuration Type

Configuration Type	9.0	9.5	9.6	9.7	9.8
Endpoints				Yes	

## My webMethods Server

### Core Functions

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					Yes
Installation					Yes
Instances					Yes
Inventory	Yes				

Function	9.0	9.5	9.6	9.7	9.8
Lifecycle	Yes				
Log access					Yes
Monitoring	Yes				
Snapshot templates		Yes			
SSO	Yes				
Status	Yes				

### Configuration Types

Configuration Type	9.0	9.5	9.6	9.7	9.8
Cluster				Yes	
Database connection	Yes				
Endpoints					Yes
Keystores		Yes			
Licenses					
Loggers					
Memory				Yes	
Ports	Yes				
Proxy				Yes	
SMTP settings		Yes			

Configuration Type	9.0	9.5	9.6	9.7	9.8
Truststores		Yes			
Users				Yes	
Variables				Yes	

## OneData

### Core Functions

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					
Installation					
Instances					
Inventory	Yes				
Lifecycle	Yes				
Log access			Yes		
Monitoring	Yes				
Snapshot templates					
SSO					
Status	Yes				

**Configuration Type**

Configuration Type	9.0	9.5	9.6	9.7	9.8
Ports	Yes				

**Optimize User Interface****Core Functions**

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					Yes
Installation					Yes
Instances					
Inventory					Yes
Lifecycle					
Log access					
Monitoring					
Snapshot templates					
SSO					
Status					Yes

**Configuration Type**

Configuration Type	9.0	9.5	9.6	9.7	9.8
Endpoints					Yes

**Platform Manager****Core Functions**

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					Yes
Installation					Yes
Instances					
Inventory	Yes				
Lifecycle	Yes				
Log access			Yes		
Monitoring	Yes				
Snapshot templates		Yes			
SSO					
Status	Yes				

## Configuration Types

Configuration Type	9.0	9.5	9.6	9.7	9.8
Authentication	Yes				
Groups	Yes				
Licenses	Yes				
Memory				Yes	
Ports	Yes				
Proxy				Yes	
Roles	Yes				
Users				Yes	

## Task Engine

### Core Functions

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					Yes
Installation					Yes
Instances					
Inventory	Yes				
Lifecycle					
Log access					Yes

Function	9.0	9.5	9.6	9.7	9.8
Monitoring					
Snapshot templates		Yes			
SSO					
Status					

### Configuration Types

Configuration Type	9.0	9.5	9.6	9.7	9.8
Endpoints					Yes

## Terracotta

### Core Functions

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					Yes
Installation					Yes
Instances					
Inventory					Yes
Lifecycle					Yes
Log access					Yes
Monitoring					Yes

Function	9.0	9.5	9.6	9.7	9.8
Snapshot templates				Yes	
SSO					
Status					Yes

### Configuration Types

Configuration Type	9.0	9.5	9.6	9.7	9.8
Cluster					Yes
Licenses					Yes
Memory					Yes
Ports					Yes
TC-CONFIG					Yes
TC-SERVER-NAME					Yes

## Universal Messaging

### Core Functions

Function	9.0	9.5	9.6	9.7	9.8
Composite templates					Yes
Installation					Yes
Instances				Yes	

Function	9.0	9.5	9.6	9.7	9.8
Inventory		Yes			
Lifecycle		Yes			
Log access				Yes	
Monitoring		Yes			
Snapshot templates		Yes			
SSO					
Status		Yes			

#### Configuration Types

Configuration Type	9.0	9.5	9.6	9.7	9.8
Cluster				Yes	
Licenses		Yes			
Memory				Yes	
Ports		Yes			
Users				Yes	