CentraSite Administrator's Guide

Version 10.5

October 2019

WEBMETHODS

This document applies to CentraSite 10.5 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.

Document ID: CS-AG-105-20210719

Table of Contents

About this Guide **5**
- Document Conventions 6
- Online Information and Support 6
- Data Protection 7

1 Configuring CentraSite **9**
- Overview 10
- Enabling JavaScript in Web Browser 10
- Starting and Stopping the Application Server Tier and Registry Repository 11
- Administering the License Key 12
- Configuring Application Server Tier 14
- Maintaining the CentraSite Internal Database 22
- Configuring High Availability 38
- Configuring Port Numbers 38
- Configuring Secure Communication Between Components 40
- Configuring Registry Cache Settings 62
- Configuring User Authentication and Repositories 65
- Configuring Email Server 90
- Implementing CentraSite 93

2 Managing Logs **101**
- Overview 102
- Configuring Logs 103
- Obtaining Log Configuration Settings 107
- Monitoring Logs 108
- Audit Logs for User Information 108
- Purging Logs 115
- Exporting the Purged Log Records 119
- Configuring Purger Properties for High Volume Data Handling 119
- Removing Leftover Auditable Events 121

3 Administering CentraSite with Command Central **123**
- Overview 124
- Viewing CentraSite Components 124
- Changing the Authentication Mode 125
- Changing the Administrator User Password for Managed Products 125
- Verifying the Outbound Authentication Settings 125
- Commands that CentraSite Application Server Tier Supports 126
- Commands that CentraSite Registry Repository Supports 126
- Lifecycle Actions for CentraSite Registry Repository 127
- Run-time Monitoring Statuses for CentraSite Registry Repository 127

4 CentraSite Command Line Tools **129**

Introduction to CentraSite Command Line Tools....130
INOADMIN Command Line Tools....130
CentraSiteCommand Line Tools....135
CentraSiteToolbox Command Line Tools....145
Configuring CentraSiteCommand to Use SSL....147

About this Guide

- Document Conventions 6
- Online Information and Support 6
- Data Protection 7

This guide describes all the configuration required for the functioning of CentraSite and the administration-level tasks that you can perform in the CentraSite environment.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format *folder.subfolder.service*, APIs, Java classes, methods, properties.
Italic	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	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 and Support

Software AG Documentation Website

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

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at https://empower.softwareag.com/.

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.

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at https://empower.softwareag.com/public_directory.asp and give us a call.

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.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 Configuring CentraSite

- Overview 10
- Enabling JavaScript in Web Browser 10
- Starting and Stopping the Application Server Tier and Registry Repository 11
- Administering the License Key 12
- Configuring Application Server Tier 14
- Maintaining the CentraSite Internal Database 22
- Configuring High Availability 38
- Configuring Port Numbers 38
- Configuring Secure Communication Between Components 40
- Configuring Registry Cache Settings 62
- Configuring User Authentication and Repositories 65
- Configuring Email Server 90
- Implementing CentraSite 93

Overview

This section describes the configuration required for setting up the CentraSite environment, such as configuring port numbers, secure communication between components, registry cache settings, user authentication and repositories, and so on. In addition, it explains various administrative tasks like administration of the internal database that hosts the registry and repository, runtime components of CentraSite, performance tuning, and error analysis.

The CentraSite metadata repository consists of the following components:

- A registry for Web services and related SOA (service oriented architecture) objects.
- A metadata repository.

Notation

In this document, the following terms are used to describe the required disk locations:

Term	Description
<Software AG_directory>	This is the root directory for all products in the webMethods suite. On Windows, this is by default C:\SoftwareAG. On UNIX, this is by default /opt/softwareag/.
<CentraSiteInstall_Directory>	This is the CentraSite installation directory. By default, this is the CentraSite folder under *Software AG_directory*.
<RuntimeDir>	This is the location of the Software AG Runtime. By default: *Software AG_directory*/profiles/CTP.
<RuntimeWebAppsDir>	This is the directory where the web applications on the Software AG Runtime are deployed. By default: *<RuntimeDir>*/workspace/webapps.
<JDKInstallDir>	This is the installation directory of the Java JDK on Windows, for example *Software AG_directory*\jvm\jvm*<n>*, where *<n>* is the JDK version number.

Enabling JavaScript in Web Browser

After installing CentraSite, you must clear your browser's cache to avoid JavaScript errors and make sure the browser is set up to allow JavaScript to execute.

Starting and Stopping the Application Server Tier and Registry Repository

The Application Server Tier is hosted in the **Software AG Runtime** that starts automatically when you start the machine on which it is installed. On Windows, you can start and stop Software AG Runtime using the Windows Services window. On UNIX, you can start and stop Software AG Runtime using these scripts:

```
<Software AG_directory>/profiles/CTP/bin/sagctp<NNN>.sh start
```

```
<Software AG_directory>/profiles/CTP/bin/sagctp<NNN>.sh stop
```

Where, `<NNN>` is the release number.

Example:

```
<Software AG_directory>/profiles/CTP/bin/sagctp100.sh start
```

The Registry Repository starts automatically when you boot your machine. On Windows, you can start or stop the Registry Repository using the Windows Services window. For other operating systems, and as an alternative method on Windows, you can start and stop the CentraSite Registry Repository from the command line.

Important:
Before using the `inoadmin` tool on Windows or UNIX systems, it is important that you run the command line script `centrasite_setenv`. This ensures that environment variables and lookup paths are set correctly for the subsequent commands.

You execute this command tool in the CentraSite Command Line Interface: `centrasite_setenv.cmd` (Windows) or `centrasite_setenv.sh` (UNIX). The tool is located in the directory `<CentraSiteInstall_Directory>/bin`.

Starting the CentraSite Registry Repository

To start the CentraSite Registry Repository, run the `inoadmin` command `start`.

The syntax is of the format: `inoadmin start <server>`

Example:

```
inoadmin start CentraSite
```

Stopping the CentraSite Registry Repository

To stop the CentraSite Registry Repository, run the `inoadmin` command `stop`.

The syntax is of the format: `inoadmin stop <server>`

Example:

```
inoadmin stop CentraSite
```

Run one of the following commands to stop the CentraSite Registry Repository:

Command	Description

`stop CentraSite`
-OR-
`stop CentraSite normal`
	This option terminates the CentraSite Registry Repository session normally and waits for any active process to finish. The waiting time in seconds can be set with the server property Maximum Transaction Duration).

`stop CentraSite rollback`	This option terminates the CentraSite Registry Repository immediately. User transactions that have not finished processing are rolled back.
`stop CentraSite abort`	This option causes an emergency shutdown of the CentraSite Registry Repository. All processing is stopped immediately. Crash dump files are written. This method causes an automatic repair (autorepair) the next time the CentraSite Registry Repository is started, and should only be used as a last resort.

You can also use Command Central to start and stop the CentraSite Registry Repository. The CentraSite Registry Repository is listed as a process in the instances list of your selected Command Central environment, and you can use standard operations in the Command Central interface to start and stop the process.

Administering the License Key

CentraSite is equipped with a license key that enables you to use the CentraSite software. The license key determines:

- Which edition of CentraSite you are licensed to use.
- The date until which your license is valid.

Relationship between the License Key and Editions

In addition to the full-feature CentraSite edition, Software AG also offers the free-of-charge CentraSite Community Edition. Your license key determines which edition is enabled for your instance of CentraSite.

If you do not specify a license key during the installation procedure, CentraSite is installed with a *default key*, which enables the Community Edition. The default key has no expiration date. If you are licensed to use the full-feature edition, you will receive an additional license key from Software AG, and you can specify this key either during the installation procedure or in a separate step after the installation procedure.

Identifying the CentraSite Edition

To determine the CentraSite edition that is running, start CentraSite Control. You can start CentraSite Control from the Windows Start menu on the machine where Software AG Runtime

is installed, or you can open a browser and type the URL: `http://Software AG Runtime_server:Software AG Runtime_port/PluggableUI/Control`

The default user ID and password are `Administrator` and `manage` respectively.

The banner at the top of the screen indicates the CentraSite edition . If the Community Edition is running, this is indicated in the banner, otherwise the full-feature edition is running.

Changing the License Key

You might wish to change the license key that CentraSite is using, for example in the following circumstances:

- Install the key that you have received from your software supplier.
- Replace an expired key.
- Switch to a different license key (for example, to upgrade to a different edition of CentraSite).

To change a license key

1. Stop the CentraSite Registry or Repository.
2. Locate the file system location where the file containing the current license key is stored. This is by default `<CentraSiteInstall_Directory>/lkey`.
3. Locate the file containing the current license key.

   This is by default `inmlicense.xml` file.
4. Rename the current license key file to a name of your choice.

   Ensure that you keep a backup copy of this file, in case you wish to revert to this license key at a later stage.
5. Copy the file containing the new license key into this file system location.

   If the name of the new file is not the same as the name you were using so far for the license key file, rename the new file to the old file name.

6. Start the CentraSite Registry or Repository.

Working with Time-limited Licenses

Certain licenses have expiration dates. If you have a time-limited license, your instance of CentraSite automatically reverts to the Community Edition when the license expires.

To check the expiration date of your license

1. Open the license key file in a text editor.
2. Locate the element `ExpirationDate`.

   If this element contains the value `Unlimited`, then there is no expiration date, that is, the use of the license is unlimited. If this element contains a date, this date is the last date for which the license is valid.

Configuring Application Server Tier

To configure the CentraSite Application Server Tier (CAST) you have the follow the following procedures:

1. Change the default Java location for CentraSite
2. Configure Java Service Wrapper
3. Configure a proxy
4. Print on UNIX and Linux systems
5. Configure UDDI

Changing the Default Java Location for CentraSite

CentraSite must point to a JDK or JRE. By default, CentraSite points to the location of the JDK (in the `<Software AG_directory>/jvm` directory) installed at the same time you installed CentraSite. You can specify a different location, if required.

You can specify a non-default JDK or JRE for CentraSite to use.

Important:
If you specify a non-default JRE or JDK, ensure that you apply maintenance updates from the appropriate vendor on a regular basis, just as you would with JREs and JDKs that you install yourself.

Before you specify the location of Java for CentraSite, determine whether you need to specify the location of the JDK or the JRE. If you intend to install CentraSite Eclipse Plug-ins to execute CentraSite reporting using BIRT, specify the location of the JDK. If you are not using this installation of CentraSite to execute BIRT reporting, you can specify the location of a JRE.

If you specify a different JDK or JRE, do not remove the JDK or JRE that Software AG Installer installed with CentraSite. The JDK and JRE installed with CentraSite are required to run the Software AG Uninstaller.

You must modify the following configuration files to point to the non-default JDK or JRE installation directory:

- `<CentraSiteInstall_Directory>/centrasite_setenv.cmd` or `centrasite_setenv.sh`
- `<Software AG_directory>/profiles/CTP/configuration/wrapper.conf`
- `<Software AG_directory>/profiles/CTP/configuration/custom_wrapper.conf`

To specify a non-default JDK or JRE

1. Open the centrasite_setenv.cmd/sh file in a text editor.

   You can find the centrasite_setenv.cmd/sh file in the directory: `<CentraSiteInstall_Directory>/bin`

   a. Set the `CS_JAVA_EXE` parameter so that it specifies the location of the JDK or JRE executable file.

      For example:

      ```
      CS_JAVA_EXE=C:\my_java\bin\java
      ```

   b. Set the `CS_JAVA_HOME` parameter so that it specifies the location of the JDK or JRE installation directory.

      For example:

      ```
      CCS_JAVA_HOME=C:\my_java
      ```

   c. Save and close the file.

2. Open the wrapper.conf file in a text editor.

   You can find the wrapper.conf file at the location: `<Software AG_directory>/profiles/CTP/configuration`

   a. Set the `wrapper.java.command` property so that it specifies the location of the non-default JDK or JRE installation directory.

      For example:

      ```
      wrapper.java.command=C:\my_java\bin\java
      ```

   b. Save and close the file.

3. Open the custom_wrapper.conf file in a text editor.

You can find the custom_wrapper.conf file at the location: <*Software AG_directory*>/profiles/CTP/configuration

a. Set the `set.JAVA_HOME` property so that it specifies the location of the non-default JDK or JRE installation directory.

   For example:

   ```
   set.JAVA_HOME=C:\my_java
   ```

b. Save and close the file.

4. Restart Software AG Runtime.

Configuring Java Service Wrapper

By default, the CentraSite installation procedure installs the CentraSite Application Server Tier on Software AG Runtime. You can configure your Software AG Runtime environment by adding or modifying Java Service Wrapper properties. The Java Service Wrapper is an application developed by Tanuki Software, Ltd. It is a utility program that launches the JVM in which CentraSite runs.

In addition, the Java Service Wrapper offers features for monitoring the JVM, logging console output, and generating thread dumps. The following sections describe how CentraSite uses the features of the Java Service Wrapper. For an overview of the Java Service Wrapper, see the webMethods cross-product document, *Software AG Infrastructure Administrator's Guide*.

The Java Service Wrapper Configuration Files

For CentraSite, the files reside in the <*Software AG_directory*>/profiles/CTP/configuration directory.

When you start Software AG Runtime, properties 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 CentraSite. *Do not modify the contents of this file unless asked to do so by Software AG.*
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, then modify this file. The settings in this file override settings in the wrapper.conf file.

The following sections describe configuration changes that CentraSite supports for Java Service Wrapper. Do not make any configuration changes to the Java Service Wrapper other than the ones described in the following sections:

JVM Configuration

When the Java Service Wrapper launches JVM, it provides configuration settings that, among other things, specify the size of the Java heap, 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 CentraSite runs.

The JVM property settings that CentraSite installs 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, *Software AG Infrastructure Administrator's Guide*.

Property	Value
`wrapper.java.initmemory`	Initial size (in MB) of the Java heap.
`wrapper.java.maxmemory`	Maximum size (in MB) to which the Java heap can grow.
`wrapper.java.classpath.` *n*	Directory in the classpath.
`wrapper.java.additional.` *n*	Java option to be passed in on the command line.

The Wrapper Log

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 CentraSite is running. The wrapper log is especially useful when you run CentraSite as a Windows service because console output is normally not available to you in this mode.

The Java Service Wrapper log for CentraSite is located in the

`<Software AG_directory>\profiles\CTP\logs\wrapper.log` file.

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 CentraSite installs 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, *Software AG Infrastructure Administrator's Guide*.

Property	Value
`wrapper.console.loglevel`	Level of messages to display in the console.
`wrapper.console.format`	Format of messages in the console.
`wrapper.logfile`	File in which to log messages.
`wrapper.logfile.loglevel`	Level of messages to write in the log file.
`wrapper.logfile.format`	Format of messages in the log file.
`wrapper.logfile.maxsize`	Maximum size to which the log can grow.
`wrapper.logfile.maxfiles`	Number of old logs to maintain.
`wrapper.syslog.loglevel`	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 CentraSite uses or allows you to configure. To learn more about these features, see the webMethods cross-product document, *Software AG Infrastructure Administrator's Guide*.

Feature	Enabled?	User configurable?
JVM timeout	Yes	No. Do not modify the `wrapper.ping` properties unless asked to do so by Software AG.
Deadlock detection	No	Yes. For more details see, Deadlock-Detecting Properties.
Console filtering	No	Yes. For more details see, Console Filtering Properties.

JVM Timeout Properties

The `wrapper.ping.interval` properties in the wrapper configuration files determine whether the wrapper monitors the JVM for timeout and what action it takes when a timeout occurs. To use timeout with CentraSite, you can configure the following properties. See the webMethods cross-product document, *Software AG Infrastructure Administrator's Guide* for procedures and additional information.

Property	Value
`wrapper.ping.interval`	How often, in seconds, the Java Service Wrapper pings JVM to ensure that it is active. The default is 5 seconds.

Property	Value
wrapper.ping.timeout	Length of time, in seconds that the Wrapper waits for a response to a ping. Set this property to 60. If the Wrapper does not receive a response in the specified time, it initiates the action specified in wrapper.ping.timeout.action.
wrapper.ping.timeout.action	Action to take if the Wrapper does not receive a response to a ping in the allotted time. Set this property to DEBUG,DUMP,RESTART.

Deadlock-Detection Properties

The wrapper.check.deadlock properties in the wrapper configuration files determine whether the wrapper monitors the JVM for deadlocks and what action it takes when a deadlock occurs. To use deadlock-detection with CentraSite, you can configure the following properties. See the webMethods cross-product document, *Software AG Infrastructure Administrator's Guide* for procedures and additional information.

Property	Value
wrapper.check.deadlock	Flag (TRUE or FALSE) that enables or disables deadlock detection. The default is FALSE.
wrapper.check.deadlock.interval	How often, in seconds, the Java Service Wrapper evaluates the JVM for a deadlock condition. The default is 60 seconds.
wrapper.check.deadlock.action	Action that occurs if the Java Service Wrapper detects a deadlock condition.
wrapper.check.deadlock.output	Information to log when the Wrapper detects a deadlock condition. Set this property to DEBUG,DUMP,RESTART.

Console Filtering Properties

The wrapper.filter properties in the wrapper configuration files determine whether the wrapper monitors the console for specified messages and what action it takes when a specified message occurs. To use console filtering with CentraSite, you can configure the following properties. However, Software AG recommends that you do not modify these properties unless asked to do so. See the webMethods cross-product document, *Software AG Infrastructure Administrator's Guide* for procedures and additional information.

Property	Value
wrapper.filter.trigger. *n*	String of text that you want to detect in the console output.

Property	Value
`wrapper.filter.action.` *n*	Action that occurs when the Java Service Wrapper detects the string of text.
`wrapper.filter.allow_wildcards.` *n*	Flag (TRUE or FALSE) that specifies whether the Java Service Wrapper processes wildcard characters that appear in `wrapper.filter.trigger.` *n*.
`wrapper.filter.message.` *n*	Message that displays when Java Service Wrapper detects the string of text.

Generating a Thread Dump

The Java Service Wrapper provides a utility for generating a thread dump of the JVM when CentraSite is running as a Windows service. A thread dump can help you locate thread contention issues that can cause thread blocks or deadlocks.

Go to the `<Software AG_directory>/profiles/CTP/bin` directory and execute the command `service -dump`. The Java Service Wrapper writes the thread dump to the wrapper.log file in the `<Software AG_directory>/profiles/CTP/logs` directory.

For information about generating a thread dump using the Java Service Wrapper, see the webMethods cross-product document, *Software AG Infrastructure Administrator's Guide*.

Configuring a Proxy

If you use a proxy to access the Internet, add the properties below to the com.softwareag.proxy.<type>.pid.properties file located at `<SoftwareAG_directory>/profiles/CTP/configuration/com.softwareag.platform.config.propsloader/`.

Create a file com.softwareag.proxy.<type>.pid.properties if it does not exist. Where <*type*> has to be replaced by either http or https.

Configuring an HTTP Proxy

To configure an HTTP proxy, add the properties below to the com.softwareag.proxy.http.pid.properties file located at `<SoftwareAG_directory>/profiles/CTP/configuration/com.softwareag.platform.config.propsloader/`.

```
type=http
host=httpprox.3rd_level_domain.2nd_level_domain
port=proxy_port_number
nonProxyHosts=*.3rd_level_domain.2nd_level_domain
|*.domain_1.extension
|*.domain_2.extension
user=proxy_user
@secure.password=proxy_password
```

Configuring an HTTPS (SSL) Proxy

To configure an HTTPS proxy, add the properties below to the com.softwareag.proxy.https.pid.properties file located at <*SoftwareAG_directory*>/profiles/CTP/configuration/com.softwareag.platform.config.propsloader/.

```
type=https
host=httpsprox.3rd_level_domain.2nd_level_domain
port=proxy_port_number
nonProxyHosts=*.3rd_level_domain.2nd_level_domain
|*.domain_1.extension
|*.domain_2.extension
user=proxy_user
@secure.password=proxy_password
```

The table below describes the properties used to configure a proxy:

Property	Description
`type`	Mandatory Specifies the type of the proxy and must be the same type as defined in the name of the configuration file. Possible values are: http, https
`host`	Mandatory Specifies the address of the proxy server.
`port`	Mandatory Specifies the address of the proxy server. The port's value must be a numeric value within the valid port range (1-65536).
`nonProxyHosts`	Specifies a pipe-separated list of host names and patterns of host names which must not be proxied through the configured proxy but accessed through a direct connection. **Note:** The `nonProxyHosts` setting should specify at least the name of the host on which the Software AG Runtime is running.
`user`	Optional Specifies the username for the proxy server when the proxy requires authentication and is omitted when authentication is not required.
`password`	Optional and must be provided only when user property is provided. Specifies the password for authenticating the user. It is not recommended to store passwords in plain text on the disc. With the proxy configuration through the configuration files the passwords can be stored with `@secure.` tokens putting them in Passman. `@secure.password=proxyPass`

Property	Description
	Having `@secure.` in front of the property key would lead to storing the property value in Passman and replacing the value with the handle pointing to the stored location in Passman.

You may need to set the proxy settings in these cases:

- To access or import WSDL files from the internet.
- In the Software AG Runtime where CentraSite Control is running.
- If you embed the importer in an application, these settings also apply.

Printing on UNIX and Linux Systems

If the Software AG Runtime is running on a UNIX or Linux system and you want to use the print feature (for example, to perform instance printing or preview BIRT reports), add the following property to the `custom_wrapper.conf` file:

```
wrapper.java.additional.<n>=-Djava.awt.headless=true
```

Configuring UDDI

You can configure the default behavior of the UDDI processing in CentraSite using global and local properties. You set these properties on each Application Server Tier in your environment, where the UDDI Registry web application runs that handles UDDI client requests to CentraSite.

Maintaining the CentraSite Internal Database

The contents of the CentraSite Registry or Repository are stored physically in an internal database. The internal database typically exists as a set of files on disk for persistent storage, together with a large memory cache during normal runtime operation.

The workings of the internal database are not revealed to end users by means of any user interface or API. Only the product administrator can access and maintain the internal database.

Repository Monitoring

While the CentraSite Registry or Repository is running, you can monitor the disk space and memory cache requirements of its internal database. Based on these values, you can optimize your installation for best use of memory and disk space.

Displaying the Database Activity

Pre-requisites:

To display the database activity information through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `showactivity` for this purpose.

The tool displays the following items of database activity information:

Item	Description
`Bufferpool size`	The total amount of memory available for caching database blocks.
`Current used bufferpool size`	The amount of memory currently being used to store cached database blocks.
`Current number of Index blocks`	The number of currently cached blocks from the Index part of the database.
`Current number of Data blocks`	The number of currently cached blocks from the Data part of the database.
`Current number of Temp blocks`	The number of currently cached blocks from the Temp part of the database.
`Number of buffer flushes`	The number of physical writes from the buffer pool to the disk that have occurred during the current CentraSite session. A buffer flush is the process whereby CentraSite copies the entire contents of the buffer pool to disk, then deletes the contents of the buffer pool. This frees up the buffer pool for subsequent logical I/O operations.
`Logical reads`	The number of database read operations that accessed the buffer pool during the current CentraSite session.
`Physical reads`	The number of database read operations that caused a physical disk access during the current CentraSite session.
`Bufferpool hit rate`	The ratio of times that a read operation was satisfied from the buffer pool rather than from the disk during the current CentraSite session, expressed as a percentage.
`Current bufferpool hit rate`	The buffer pool hit rate, limited to the time period between the previous and current activation of the command `showactivity`.
`Flush limit`	The maximum amount of buffer pool space that can be in use before a buffer flush takes place. If this value is exceeded, an automatic buffer flush occurs.
`Modified pages in bufferpool`	The ratio of modified pages to unmodified pages in the buffer pool, expressed as a percentage.
`Dynamic pool size`	The size of the dynamic pool. The dynamic pool is a separate cache area, not part of the main buffer pool, and is used as a work area for

Item	Description
	certain operations such as sort and search operations. The dynamic pool is shared across all users of the system.
`Current used dynamic pool size`	The amount of the dynamic pool currently in use.
`Maximum pool usage`	The high-water mark of the dynamic pool usage during the current session.
`Current number of space waiters`	The number of users or applications that are currently waiting for space allocation in the dynamic pool.
`Total number of space waiters`	The total number of users or applications that have waited for space allocation in the dynamic pool during the current session.

To display the database activity information

- Run the command `showactivity`.

  The syntax is of the format: `inoadmin showactivity <server>`

  Example:

  ```
  inoadmin showactivity CentraSite
  ```

Displaying the Database Space

To display the database space information through the CentraSite Command Line (.cmd) Interface, you must have the CentraSite Administrator role.

You can display information about the physical disk requirements of CentraSite's internal database. The database consists of several components called *spaces*, and each database space is stored in its own physical file. There are several kinds of database space:

Database space	Description
`Data`	Contains the data of the CentraSite Registry or Repository.
	The file type of a data space is `2D0`. An example of a file name is `AAB00001.2D0`.
`Index`	Contains the indexes that CentraSite maintains for retrieving stored data.
	The file type of an index space is `2I0`. An example of a file name is `AAB00001.2I0`.
`Journal`	Contains information required for rolling back transactions.

Database space	Description
	The file type of a journal space is `2J0`. An example of a file name is `AAB00001.2J0`.
`Log`	Contains a log of all database modifications that have occurred in the current CentraSite session. The file type of a log space is `2L0`. An example of a file name is `AAB000010000000052.2L0`. The name is composed of the filename of the database's data and index spaces (for example, `AAB00001`), followed by a sequence number. The sequence number is incremented by 1 for each new log space.
`Backup`	Contains a backup of the CentraSite Registry Repository. There is an initial backup The file type of a backup space is `2B0`. An example of a file name is `AAB00001001334723430.2B0`. The name is composed of the filename of the database's data and index spaces (for example, `AAB00001`), followed by an integer that represents the date and time when the backup was created.

CentraSite provides a command tool named `listdbspaces` for this purpose.

The tool displays the following information for each of the database space:

- The type of database space, for example, Data, Index, Journal, Log or Backup. There can be several spaces of the same type.
- The amount of disk storage that this database space uses.
- The location of the database space in the file system and the name of the physical file that contains the database space.

To display the database space information

- Run the command `listdbspaces`.

  The syntax is of the format: `inoadmin listdbspaces <server>`

  Example (all in one line):

  ```
  inoadmin listdbspaces CentraSite
  ```

Displaying Backup List

To view the list of available backups through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

You can view the list of all available backups and their backup keys (the unique identifiers for the backups) in CentraSite.

CentraSite provides a command tool named `listbackups` for this purpose.

To display the list of available backups

- Run the command `listbackups`.

  The syntax is of the format: `inoadmin listbackups <server>`

  Example:

  ```
  inoadmin listbackups CentraSite
  ```

Backing Up the Database

Pre-requisites:

To back up the database information through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

To protect against accidental loss of data, **Software AG** recommends that you take regular backups of the internal database in which CentraSite's registry and repository data is stored. When you make a backup, you copy the contents of the internal database to a file on the file system. At a later stage, you can retrieve the contents of a backup and restore them into the internal database.

CentraSite provides a command tool named `backup` for this purpose.

When you create a backup, the backup file is stored by default in the predefined location for backups, but you can optionally specify a different backup location. For more information about predefined locations, see "Configuring CentraSite Database Locations" on page 31

Note:
During the installation of CentraSite, a backup of the initial database state is automatically created, with a timestamp equal to the date and time when you install the product. If you for any reason wish to restore the database to its initial state, that is, the state immediately after product installation, you can use this backup.

You must decide whether you want to create the backup in the default backup location or in a different location. If you are choosing a different location, you must specify a location that already exists, otherwise the backup will not run.

To back up the internal database

- To create the backup in the default backup location, run the command `backup`.

  The syntax is of the format: `inoadmin backup <server>`

  Example (all in one line):

  ```
  inoadmin backup CentraSite
  ```

- To create the backup in a location other than the default backup location, run the command `backup` with `<Location>` parameter.

The syntax is of the format: `inoadmin backup <server> <Location>`

where, `<Location>` is the path where the backup will be created.

Example (all in one line):

```
inoadmin backup CentraSite C:\SoftwareAG\AnotherBackupLocation
```

The Backup Key

When the backup completes, a status message appears on the command line, indicating the date and time when the backup was created. In addition, a backup key is displayed. The backup key is a unique identifier for the backup. If you wish to restore the backup at a later stage, you identify the backup using the backup key.

Restoring the Database from a Backup

Pre-requisites:

To restore the database backup through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

You can change the contents of CentraSite's internal database back to a previous state by restoring a backup. When you restore a backup, you completely replace the current contents of the database by the contents that existed when the backup was made.

CentraSite provides a set of command tools for this purpose.

Repository changes that are made between one backup and the subsequent backup are stored in session logs. When you restore from a backup, you can optionally select to include or omit the data from the session logs.

As soon as the restore step successfully finishes, the repository is automatically started in standby mode. Then the recover step is started, in which all changes that were made since the last backup are reapplied from the session logs. Finally, the repository is shut down again. The restore function can only be used when the repository is inactive (stopped).

Note:
During the installation of CentraSite, a backup of the initial repository state is automatically created, with a timestamp equal to the date and time when you install the product. If you for any reason wish to restore the repository to its initial state, that is, the state immediately after product installation, you can use this backup.

To restore the database from a backup

1. First, back up the current database, in case you decide to return to this database state at a later time.

2. Stop the CentraSite Registry or Repository.

3. To show a list of all available backups and their backup keys (the unique identifiers for the backups), run the command `listbackups`.

   The syntax is of the format `inoadmin listbackups <server>`

4. Identify the backup file you wish to use for the restore operation.

5. To restore from the selected backup, run the command `restore`.

   The syntax is of the format `inoadmin restore <server> <BackupKey> <RecoverOption>`

6. Select the recover option that you want to use. If you do not specify a recover option, this has the same effect as using the option `recover all`.

   The following recover options are available:

Option	Action
`recover all`	The database is restored to the state stored in the backup, and all session logs created since the backup are included. This is the default option.
`recover no`	The database is restored to the state stored in the backup. No session log data is processed.
`recover <UntilDateTime>`	All session logs created after the specified time and date are excluded from the recovery. The format of this field is: `DD-MMM-YYYY:HH:MM:SS` The month is given as a 3-letter abbreviation, using the first 3 characters of the month's name. Example: 23-OCT-2012:16:52:30
`recover`	This has the same effect as `recover all`.

   The following example restores the database from a backup whose backup key is `001334732430`. All database changes that occurred after the backup was made, up to and including 11:30am on October 25, 2012, is also retrieved from the session logs.

```
inoadmin restore CentraSite 001334732430 recover 25-OCT-2012:11:30:00
```

Disabled Backups

If you select to restore a backup without full recovery by using the recovery option `no`, and there are one or more backups that are more recent than the backup being restored, these more recent backups are set to disabled.

If you select to restore a backup without full recovery by using the option `recover` <UntilDateTime> , and there are one or more backups that are more recent than the time specified by this option, these more recent backups are set to disabled.

A disabled backup cannot be accessed any more from any of the CentraSite user interfaces, and in particular cannot be accessed for restore or recover operations. It is not displayed in the list of available backups.

Disabled backups remain on the backup location on your disk as long as there are older, non-disabled backups. If you delete all non-disabled backups that are older than a given disabled backup, CentraSite automatically deletes the disabled backup.

If you should wish to reactivate a disabled backup, archive it using standard operating system functionality and contact your software supplier.

Moving a database / Disaster Recovery

Under normal circumstances, a database backup that is created on one machine can only be restored to the same machine. However, a database backup that originated on one machine can be configured, so that it can be restored onto another machine. For more information, see "Moving a Database to Another Location" on page 30.

Deleting a Backup

Pre-requisites:

To delete a backup through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

You can delete backups that are no longer required. Deleting a backup removes all the backup spaces that are associated with it, but the associated session log information is not removed, since it may subsequently be required if the database has to be recovered.

CentraSite provides a command tool named `deletebackup` for this purpose.

To delete a backup

1. Listing available backups and backup keys: Run the command `listbackups`.

   The syntax is of the format `inoadmin listbackups <server>`

2. Identify the backup that you wish to delete.

3. Deleting a Backup: Run the command `deletebackup`.

   The syntax is of the format: `inoadmin deletebackup <server> <BackupKey>`

   where `<BackupKey>` is the backup key of the backup you wish to delete.

   Example:

   ```
   inoadmin deletebackup CentraSite 001334732430
   ```

Post-requisites: If, after you delete a backup, there are one or more disabled backups older than the oldest remaining non-disabled backup, CentraSite automatically deletes all of these older disabled backups.

Moving a Database to Another Location

Pre-requisites:

To move a CentraSite database through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

Under certain circumstances you might wish to move a CentraSite database from an existing CentraSite installation into another CentraSite installation, either on the same machine or on a different machine. Examples of such situations are:

- When resources for processing become insufficient.
- In a disaster recovery scenario.
- As part of a side-by-side product installation, whereby two versions of CentraSite can exist on the same machine.

Moving a CentraSite database to another machine is not supported in the following scenarios:

- if the architectures of the source and the target machine is with different byte orders (big-endian versus little-endian)
- if the source machine is a UNIX system and the target machine is a Windows system.

If you are not sure whether moving a database is supported in your environment, contact a technical representative of Software AG.

The CentraSite registry or repository contains environment-dependent data. This data has to be adjusted to the new environment when the data is moved to another machine.

CentraSite provides a command tool named `MoveCentraSiteRR` for this purpose. The tool allows you to create a new database from an existing backup and to modify the data to the needs of the new environment.

The steps performed by the command tool are:

- Calculate database space sizes
- Delete the existing CentraSite database
- Create a new CentraSite database
- Adjust the environment specific data
- Create a new backup containing the modifications

Important:
The contents of the backup file overwrites the database of the target CentraSite installation.

To move the database to another location

- Run the command `MoveCentraSiteRR`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\bin\cfg>MoveCentraSiteRR.cmd backup-filename [data-space-location]`

The input parameters are:

Parameter	Description
`backup-filename`	(Optional). Name of the existing backup file.
`data-space-location`	The absolute path to the CentraSite database. Default value is C:\SoftwareAG\CentraSite\data.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\bin\cfg>MoveCentraSiteRR.cmd
C:\SoftwareAG\CentraSite\data\AAB00001001461149722.2B0 C:\SOAPlatform\CentraSite\data
```

Configuring CentraSite Database Locations

To configure the database locations of CentraSite through the CentraSite Command Line (.cmd) Interface, you must have the CentraSite Administrator role.

CentraSite uses certain default locations on disk to store information about the active CentraSite Registry or Repository session and the backup environment. These locations are:

Location type	Purpose	Default path
`temporary`	The Temporary Working Location. This location contains temporary data that is required during normal database operation.	`<CentraSiteInstall_Directory>\data`
`backup`	The Backup Location. This is the location where backup files are created by default.	`<CentraSiteInstall_Directory>\data`
`log`	The Log Location. This is the location where session log files are created by default. If the log archive location is defined, the log location holds only the most recent session log, and all other logs are stored in the log archive location.	`<CentraSiteInstall_Directory>\data`
`archive`	The Log Archive Location. This is the location where all session logs other than the current session log are stored. If no log archive location is defined, all session log files are held in the log location,	(no default defined)

Location type	Purpose	Default path
	regardless of whether they are the log file of the current session or a previous session.	
`reserved`	The Reserved Location. This location is used as an overflow area if the database's standard locations have filled up.	`<CentraSiteInstall_Directory>\data`

The CentraSite installation procedure defines default paths for each of these location types. Depending on your storage requirements, you can change these defaults to use different or additional paths.

CentraSite provides a set of `inoadmin` command tools for monitoring and configuring the database locations.

To configure the CentraSite database locations

- Listing the Currently Defined Locations: To list the available locations on disk, run the command `showlocations`.

  The syntax is of the format: `inoadmin showlocations <server>`

  Example:

  ```
  inoadmin showlocations CentraSite
  ```

- Changing the Currently Defined Path of a Location: To change the currently defined path of a location, run the command `setlocation`.

  The syntax is of the format: `inoadmin setlocation <server> <`*`LocationType`*`> <`*`path`*`> [<`*`path`*`> ...]`

  where <*LocationType*> is one of the location types. The location can be assigned to more than one path. If several paths are defined, they are used in the order specified from left to right, when there is no more disk space available in a path, the next path is used.

  If the path contains spaces or characters that the operating system treats as escape characters in the command line (for example, the backslash character `\` in Windows), you must enclose the path in quotes.

  The path or paths you specify completely replace the previously specified path or paths. If you want to extend the current path specification with an additional path, you need to use the `setlocation` command and specify both the existing path and the new path.

  The following example sets the new default path for the Backup location to `C:\backuploc1` and `D:\backuploc2`. This means that `C:\backuploc1` is used as long as there is available disk space, if it is full, then `D:\backuploc2` is used instead.

  ```
  inoadmin setlocation CentraSite backup C:\backuploc1 D:\backuploc2
  ```

Database Configuration Parameters

You can configure various properties of CentraSite's internal database. The available properties are:

Property	Description
`buffer pool size`	This defines the size of the buffer pool that is used for storing intermediate results during normal processing.
`maximum transaction duration`	This defines the maximum time, in seconds, that a transaction is allowed to exist.
`non-activity timeout`	This defines the maximum time, in seconds, that a transaction is allowed to be inactive.
`XML work threads`	This defines the number of XML-processing threads that be active concurrently in the internal processing engine.
`XML maximum sessions`	This defines the maximum number of user sessions that CentraSite can process concurrently.
`number of backup generations`	The number of full backups that CentraSite keeps in parallel. When this number is exceeded, the oldest backup and the corresponding log spaces are deleted.
`write_limit`	The amount of the modified buffer pool space that triggers a flush (disk write) of the modifications present in the buffer pool. The default for the buffer pool size is 60MB. If this property is set to 0, CentraSite adjusts the flush limit automatically.
`replication delay`	The time interval (in seconds) after which a replication server is updated. This property is set for a master server and it applies to all replication servers that are defined for the master server.

For each property, the following information is available:

Property	Description
`handle`	A unique identifier used for internal purposes.
`minimum`	The minimum allowed value that can be configured for this property.
`maximum`	The maximum allowed value that can be configured for this property.
`default`	The value of the property that is to be used if not set explicitly to a specific value.
`configured`	The value of the property that is to be used when the server is restarted.
`current`	The value of the property currently used by the active server.

Property	Description
`type`	The data type of the property. For example, string, numeric, and so on.
`unit`	The unit of measurement for the property's value. For example, megabytes or seconds.
`state`	The state of the server. For example, active or inactive.

To examine or modify the database properties, you can run one of the following `inoadmin` commands in the `bin` folder in the *<CentraSiteInstall_Directory>*.

Command	Description
`listproperties`	Show a list of all of the available database properties and their values. The information displayed includes (where appropriate) the property name, the maximum and minimum allowed values, the current value, the configured value, and the high water mark.
`getproperty <server> <PropertyName>`	Show the value of the given property. If the property name contains one or more spaces, enclose the name in quotes.
`setproperty <server> <PropertyName> <PropertyValue> [no]restart`	Modify the value of the given property to the given value. If the property name or property value contains one or more spaces, enclose the name or value in quotes. The changed value takes effect at the next restart of the CentraSite Registry or Repository. You can cause an automatic restart by specifying `restart`, otherwise specify `norestart`.
`setproperties <server> <XMLInputFile> [no]restart`	Modify the values of several properties to the values specified in the supplied XML file. The XML file must use the same element structure as the XML file created by the `listproperties` command. The modified values takes effect at the next restart of the CentraSite Registry or Repository. You can cause an automatic

Command	Description
	restart by specifying `restart`, otherwise specify `norestart`.

Reclaiming Disk Space in CentraSite Database

To reclaim space in the CentraSite internal database through the CentraSite Command Line (.cmd) Interface, you must have the CentraSite Administrator role.

Within the CentraSite internal database a considerable amount of data may be stored temporarily, for example, log data that is purged regularly or metrics data stored by other webMethods sub systems. When such data is stored the space required for the database increases. However, after deleting such data the space allocated remains. Although the allocated space is re-used when additional data of the same type is stored again, the administrator may prefer to reorganize the database so that unused space can be reclaimed.

CentraSite provides a command tool named `reorganize` for this purpose.

The tool reduces the disk space by returning blocks that are no longer used to the file system.

The reorganize function uses a temporary backup to defragment the free space. Thus it is a pre-requisite that the space required for one CentraSite backup is available on the backup location. This temporary backup is deleted when the reorganize function completes successfully.

Important:
The command tool can be used only on an inactive database. Attempting to reorganize an active database is not allowed.

To reclaim database space

1. Stop the database you want to reorganize.

2. Ensure that there is enough space on the database backup location to hold one full backup of the CentraSite database. This space is required because the `reorganize` command creates a temporary backup while it defragments the database. The temporary backup is deleted when the reorganize command completes successfully.

3. To check the available free disk space, run the command `reorganize` with `evaluate` parameter.

   The syntax is of the format: `inoadmin reorganize <server> evaluate`

   Example:

   ```
   inoadmin reorganize CentraSite evaluate
   ```

   The output from this command shows the current amount of free disk space.

   ```
   INODSI1183: 161.84 MB empty data space found.
   ```

4. To start database reorganization, run the command `reorganize`.

   The syntax is of the format: `inoadmin reorganize <server>`

   Example:

   ```
   inoadmin reorganize CentraSite
   ```

5. If the `reorganize` command completes successfully, restart the database.

Setting Up Replication Instances of Registry Repository

To replicate instances of the CentraSite Registry Repository through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

Replication instances of the CentraSite Registry Repository (CRR) can be set up on the base of a backup from the original instance, also called master instance.

To manage the replication instance of Registry Repository, CentraSite provides a set of `inoadmin` command tools.

To be able to replicate instances of the CentraSite Registry Repository, note the following points:

- A replication instance must be configured on a different host than the master host.
- The master server must be reachable in the same domain as the replication host.
- The master server must have enough log space data available to supply the replication instances with replication data.

  The latter is important in case of a larger replication delay due to a disconnected master server. The sets of log spaces are associated with backups.

To ensure that replication instances can be supplied with replication data, the server property `number of backup generations` must be at least 2 or even larger. Once a replication instance is configured, the replication instance is started in the same way as the master server. During startup, the replication server connects to the master server to request replication data. When the connection to the master server is interrupted, the replication server re-connects as soon as the master server is reachable. If the replication server must be shut down for any reason, the replication processing is suspended and replication processing resumes when the replication server is started again.

To prepare a server to be set up as replication server, you must create a copy of the CentraSite Registry Repository using the command tool `MoveCentraSiteRR`. For details about the usage of command tool, see "Moving a Database to Another Location" on page 30.

You must not start the server of the created instance before it is set as replication instance using the command tool `setreplication`.

Note:
You must specify the host name of the computer where the CentraSite Registry Repository component is running, without specifying the domain name.

- Enabling CRR Database for Replication: To register the host name of the replication instance, at the master host, run the command `allowreplication`.

  The syntax is of the format: `inoadmin allowreplication <server> <replication_hostname>`

  This is required for a replication server that connects to a master server to request replication data.

  CentraSite does not allow a replication server to request replication data from the master server if the host name is not already registered for replication.

- Setting Up Replication Database: To set a copy of the CentraSite Registry Repository as replication instance of the master host, at the replication host, run the command `setreplication`.

  The syntax is of the format: `inoadmin setreplication <server> <master_hostname> <master_port>`

  A request for replication permission is sent to the master server on the specified host using the given port number. If the response to this request is returned successfully, the host name and port number for master server access is stored. For more information about the replication properties, see "Database Configuration Parameters" on page 33.

  Note:
  If a master server was not restarted after the first replication host name was added, a restart of the master sever may be required to change some initial set up for a running server.

- Re-setting Replication Database: To reset the replication instance to a normal server instance which can be used instead of the master instance in case of a failure, at the replication host, run the command `resetreplication`.

  The syntax is of the format: `inoadmin resetreplication <server>`

  Other replication instances cannot use such a reset server instance as the new master because of synchronization reasons. To use such an instance as a master server, a new backup should be created as a base for re-creating other replication server instances.

- Fetching Details of Replication Database: To check whether the server is a master server or a replication serve, master host or replication host, run the command `replicationinfo`.

  The syntax is of the format: `inoadmin replicationinfo <server>`

  - For a master server instance, the list of permitted replication host names is displayed as an additional information, if the master server is running.
  - For a replication server instance, the master server host and the port used for the master server access are displayed as an additional information. The replication time stamp shows the time of the last replicated data.

- Disabling CRR Database for Replication: To unregister the host name of the replication instance to prevent further re-replication, at the master host, run the command `allowreplication`.

  The syntax is of the format: `inoadmin denyreplication <server> <replication_hostname>`

Configuring High Availability

Multiple instances of CentraSite can be clustered together to provide high availability (HA). For information about configuring CentraSite for high-availability, refer to the following CentraSite HA documents located in the *files/ha/common folder* under the CentraSite installation directory:

- **CentraSiteFailoverBasics.pdf**
- **CentraSiteHASampleScript.pdf**
- **SettingUpaFailoverSolutionForCentraSite.pdf**

Configuring Port Numbers

This topic gives information about HTTP or HTTPS port numbers used by CentraSite components. In general there is no need to change these values, unless your site's requirements differ from the CentraSite default values.

Changing Port Numbers of CentraSite Registry Repository

The CentraSite Registry Repository uses the following port number that has the default value:

Port Name	Description	Default port number
HTTP port	The HTTP Server port. The TCP/IP port used for HTTP access to the CentraSite Registry Repository.	53313

To change the default port number, the following locations must be updated:

- The registry of the machine on which the CentraSite Registry or Repository is installed.
- The CAST web applications (only if the HTTP port is updated).

Important:
To avoid inconsistencies, it is important to modify the port numbers in all locations in a single step.

Changing Port Number on CentraSite Registry Repository

Pre-requisites:

To change the port number of the CentraSite Registry Repository (CRR) through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a set of command tools for this purpose.

To change a configured CRR port on the Registry Repository host

1. To ensure that environment variables and lookup paths are set correctly, run the command `centrasite_setenv`.

   The syntax is of the format: `C:\SoftwareAG\CentraSite\bin>centrasite_setenv`

2. To check the current value of the port number, run the command `inoadmin getproperty CentraSite`.

   The syntax is of the format: `inoadmin getproperty CentraSite <ParameterName>`

   The `<ParameterName>` can be any of the port names.

   Example:

   ```
   inoadmin getproperty CentraSite "HTTP port"
   ```

3. To assign a new port number, run the command `inoadmin setproperty CentraSite`.

   The syntax is of the format: `inoadmin setproperty CentraSite <ParameterName> <NewPortNumber> norestart`

   Where, `<NewPortNumber>` is the new port number that you want to use.

4. Stop the CentraSite Registry Repository.

5. Restart the CentraSite Registry Repository.

Changing Software AG Runtime Port Numbers

The default Software AG Runtime port numbers for the CAST components are 53307 for plain HTTP communication and 53308 for HTTPS communication.

The port numbers are configured in property files that are located in `<Software AG_directory>/profiles/CTP/configuration/com.softwareag.platform.config.propsloader`. If for any reason these port numbers are unsuitable (for example, your environment might require these port numbers for a non-Software AG application), you can change them in the appropriate property files as follows:

- `com.softwareag.catalina.connector.http.pid-CentraSite.properties`: This file contains parameter settings for HTTP communication.
- `com.softwareag.catalina.connector.https.pid-CentraSite.properties`: This file contains parameter settings for HTTPS communication.
- `com.softwareag.catalina.connector.http.pid-CentraSite-CrrHttp.properties`: This file is provided for compatibility with previous product releases, for which different default port numbers were used.

To change the Software AG Runtime port numbers

1. Open the property file in a rich text editor.

   You can find the property files in the `<Software AG_directory>/profiles/CTP/configuration/com.softwareag.platform.config.propsloader` directory.

2. Set the value of the new port number.

3. Restart Software AG Runtime.

Configuring Secure Communication Between Components

This section gives information about how to set up secure communication between CentraSite components, on the basis of SSL.

If you change the default configuration, you might also need to modify other products based on CentraSite. Changing the CAST configuration can affect applications such as:

- Clients that use the CAST web applications.
- Web services deployed by CentraSite-enabled products.

Securing Communication Between the CRR and the CAST

The communication between the CRR and the CAST components takes place using the 2-way SSL authentication. For this full client/server SSL communication, the client and server must accept each other's certificates. This means that the CAST and CRR stores need to have matching certificates for the communication to work.

The CAST components have access to an SSL context to establish an SSL (HTTPS) connection to the CRR. The SSL authentication establishes a trusted relationship between the CentraSite Server on the CAST and the CRR. Therefore no user re-authentication needs to be performed by the CRR.

The CentraSite installation comes with self-signed certificates from Software AG.

You can configure a secure communication between the CRR and CAST. CentraSite provides a set of command line tools for this purpose.

Note:
Keep in mind that you must run the command tool on the machine hosting an CAST or CRR environment.

You can disable the SSL communication between the CRR and the CAST components. However, Software AG strongly recommends you not to do this, because it opens a potential security risk.

Obtaining Security Configuration of CentraSite Registry Repository

Pre-requisites:

To fetch the security communication for CentraSite Registry Repository (CRR) through the Command Line Interface, you must have the CentraSite Administrator role.

You can view the SSL security values of CentraSite Registry Repository environment.

CentraSite provides a command tool named `get SSL RR` for this purpose.

- Run the command `get SSL RR`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get SSL RR -file <CONFIG-FILE>`

  The input parameters are:

Parameter	Description
`CONFIG-FILE`	The absolute or relative path to the XML configuration file, `RR-config.xml`, containing the security properties. If relative, the path should be relative to the location from where the command is executed.

  Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get SSL RR -file RR-config.xml
```

  The response to this command is as follows:

```
Executing the command : get SSL RR
Successfully executed the command : get SSL RR
```

  Sample RR-config.xml configuration file is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
  <properties>
    <entry key="com.softwareag.centrasite.security.keyStore">
      C:/SoftwareAG/CentraSite/test/files/certs/castcert.p12
    </entry>
    <entry key="com.softwareag.centrasite.security.keyStorePassword">
      cscert
    </entry>
    <entry key="com.softwareag.centrasite.security.keyStoreType">PKCS12
    </entry>
    <entry key="com.softwareag.centrasite.security.trustStore">
      C:/SoftwareAG/CentraSite/test/files/certs/casttrust.p12
    </entry>
    <entry key="com.softwareag.centrasite.security.trustStorePassword">
      cscert
    </entry>
    <entry key="com.softwareag.centrasite.security.trustStoreType">
      PKCS12
    </entry>
    <entry key="com.softwareag.centrasite.security.crr.trustStore">
      C:/SoftwareAG/CentraSite/test/files/certs/crrtrust.pem
    </entry>
    <entry key="com.softwareag.centrasite.security.crr.certificate">
      C:/SoftwareAG/CentraSite/test/files/certs/crrcert.crt
    </entry>
    <entry key="com.softwareag.centrasite.security.crr.keyFile">
      C:/SoftwareAG/CentraSite/test/files/certs/crr.key
    </entry>
```

```
    <entry key="com.softwareag.centrasite.security.crr.storePassword">
      cscert
    </entry>
  </properties>
```

The RR configuration file has two sets of SSL parameters, `com.softwareag.centrasite.security.*` and `com.softwareag.centrasite.security.crr.*`. The `com.softwareag.centrasite.security.crr.*` properties enables the SSL communication for CRR Java-based server extensions and the `com.softwareag.centrasite.security.*` properties enables the SSL CAST communication.

Note:
The `crr.storePassword` is only needed if the `com.softwareag.centrasite.security.crr.keyFile` private key file is encrypted.

Setting Security Configuration for CentraSite Registry Repository

To configure the secure communication for CentraSite Registry Repository (CRR) through the Command Line Interface, you must have the CentraSite Administrator privileges.

To define the SSL security values for use in the CentraSite Registry Repository from the command line, you must perform the following high-level steps:

- Export the RR configuration file (`RR-config.xml`) to a editable format using the `get SSL RR` command.
- Modify the SSL RR configuration parameters.
- Execute the `set SSL RR` command to define the SSL security values for CRR.

To set the security configuration for registry repository

1. Export the RR configuration file (`RR-config.xml`) to an editable format. For more information, see "Obtaining Security Configuration of CentraSite Registry Repository" on page 40.

   The `com.softwareag.centrasite.security.crr.*` properties must be in basic non-store formats, PEM key or cert or cacert.

   The default configuration, the same CA certificate is used for both client and server certificates.

   To update the SSL security values of the port, modify the `com.softwareag.centrasite.security.crr.*` properties.

2. To define the SSL security values for CRR, run the command `set SSL RR`.

   Note:
   You have to run the `set SSL RR` command as a windows administrator.

   The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set SSL RR -url <CENTRASITE-URL> -user <USER-ID> -password <PASSWORD> -file <CONFIG-FILE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	The URL of the CentraSite Registry Repository (CRR). For example, https://localhost:53313/CentraSite/CentraSite.
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, Administrator.
PASSWORD	The password for the registered CentraSite user identified by the parameter USER-ID.
CONFIG-FILE	The absolute or relative path to the XML configuration file, RR-config.xml, containing the security properties. If relative, the path should be relative to the location from where the command is executed.

Note:
If you change the default configuration, this command modifies the SSL configuration for RR. A time stamped archive of the previous configuration is available in the configuration file crr-config.YYYY-MM-DD_HH-MM-SS.xml in the folder *<CentraSiteInstall_Directory>*/cfg/archive.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set SSL RR -url
https://localhost:53313/CentraSite/CentraSite -user Administrator -password manage
-file RR-config.xml
```

The response to this command is as follows:

```
Executing the command : set SSL RR
Successfully executed the command : set SSL RR
```

Obtaining Security Configuration of CentraSite Application Server Tier

Pre-requisites:

To fetch the security communication for CentraSite Application Server Tier (CAST) through the Command Line Interface, you must have the CentraSite Administrator role.

You can view the SSL security values of CentraSite Application Server Tier environment.

CentraSite provides a command tool named `get SSL AST` for this purpose.

- Run the command `get SSL AST`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get SSL AST -file <CONFIG-FILE>`

  The input parameters are:

Parameter	Description
`CONFIG-FILE`	The absolute or relative path to the XML configuration file, `AST-config.xml`, containing the security properties. If relative, the path should be relative to the location from where the command is executed.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get SSL AST -file
AST-config.xml
```

The response to this command is as follows:

```
Executing the command : get SSL AST
Successfully executed the command : get SSL AST
```

Sample AST-config.xml configuration file is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
  <properties>
    <entry key="com.softwareag.centrasite.security.keyStore">
      C:/SoftwareAG/CentraSite/test/files/certs/castcert.p12
    </entry>
    <entry key="com.softwareag.centrasite.security.keyStorePassword">
      cscert
    </entry>
    <entry key="com.softwareag.centrasite.security.keyStoreType">
      PKCS12
    </entry>
    <entry key="com.softwareag.centrasite.security.trustStore">
      C:/SoftwareAG/CentraSite/test/files/certs/casttrust.p12
    </entry>
    <entry key="com.softwareag.centrasite.security.trustStorePassword">
      cscert
    </entry>
    <entry key="com.softwareag.centrasite.security.trustStoreType">
      PKCS12
    </entry>
  </properties>
```

Setting Security Configuration for CentraSite Application Server Tier Components

To configure the secure communication for CentraSite Application Server Tier (CAST) through the Command Line Interface, you must have the CentraSite Administrator privileges.

You can define the SSL security values for use in the CentraSite Application Server Tier using the `set SSL AST` command. To define the SSL security values for CentraSite Application Server Tier to connect to CRR, you must perform the following high-level steps:

- Export the AST configuration file (`AST-config.xml`) to a editable format using the `get SSL AST` command.
- Modify the SSL AST configuration parameters.

- Execute the `set SSL AST` command to define the SSL security values for AST.

To set the security configuration for CentraSite application server tier components (CAST)

1. To export the AST configuration file (`AST-config.xml`) to an editable format. For more information, see "Obtaining Security Configuration of CentraSite Application Server Tier " on page 43.

   To update the SSL settings for outbound AST traffic, modify the `com.softwareag.centrasite.security.*` properties.

   Note:
   When AST and RR components are authenticated with the 2-way SSL environment, the authentication does not work if the security configuration of one of the components AST or RR is modified. So if you intend to modify the default security configuration, ensure that you modify the configuration for both components AST and CRR. In addition, ensure that you execute the `set SSL RR` command before you execute the `set SSL AST` command.

2. To define the SSL security values for CAST, run the command `set SSL AST`.

   The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set SSL AST -url <CENTRASITE-RR-URL> -user <USER-ID> -password <PASSWORD> -file <CONFIG-FILE>`

   The input parameters are:

Parameter	Description
`-url`	The URL of the CentraSite registry. For example, `https://localhost:53313/CentraSite/CentraSite`. **Note:** Even when you are setting the CAST properties, you need to reference the RR port in the URL.
`-user`	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, `Administrator`.
`-password`	The password for the registered CentraSite user identified by the parameter `-user`.
`-file`	The absolute or relative path to the XML configuration file, `AST-config.xml`, containing the security properties. If relative, the path should be relative to the location from where the command is executed.

   Note:
   If you change the default configuration, this command modifies the SSL configuration for RR. A time stamped archive of the previous configuration will be available in the configuration file `cast-config.YYYY-MM-DD_HH-MM-SS.xml` in the folder `<CentraSiteInstall_Directory>/cfg/archive`.

Example:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set SSL AST -url
https://localhost:53313/CentraSite/CentraSite -user Administrator -password manage
-file AST-config.xml
```

The response to this command is as follows:

```
Executing the command : set SSL AST
Successfully executed the command : set SSL AST
```

CAST Stores

The CentraSite installation comes with self-signed certificates from Software AG. These are:

- The keystore certificate, which is located in `<CentraSiteInstall_Directory>/cast/files/certs/castcert.p12`. It contains the client certificate and private client key.
- The truststore certificate, which is located in `<CentraSiteInstall_Directory>/cast/files/certs/casttrust.p12`. This would normally contain the server certificate but actually contains the CA certificate and key.

These files need to be in a Java readable format.

Note that in the default configuration, the same CA certificate is used for both client and server certificates.

Identifying the Communication Method Between CAST and CRR

Pre-requisites:

To identify the communication method between the CentraSite Application Server Tier (CAST) and the CentraSite Registry or Repository (CRR) from the Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `get SSL usage` for this purpose.

To identify the communication method

- Run the command `get SSL usage`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get SSL usage`

  Note:
  The command output shows HTTPS if SSL usage is enabled; and HTTP if disabled.

Allowing HTTP Communication Between CAST and CRR

Pre-requisites:

To enable the HTTP communication between CentraSite Application Server Tier (CAST) and CentraSite Registry or Repository (CRR) through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

It is possible to change the communication between CAST and CRR from full 2-way SSL (HTTPS) communication to HTTP communication.

CentraSite provides a command tool named `set SSL usage` for this purpose.

CAUTION: Software AG strongly advises you to use 2-way SSL at all times for this communication. If you intend to use HTTP rather than HTTPS communication, consider carefully as using HTTP communication raises a potential security risk.

Some internal communication between CAST and CRR must always use SSL, therefore you cannot switch off HTTPS altogether.

Before you change the SSL communication between CAST and CRR, make sure you use the `inoadmin setproperty CentraSite` command to change the communication method setting as follows:

The syntax is of the format: `inoadmin setproperty CentraSite "communication method" "HTTP and HTTPS" restart`

- Run the command `set SSL usage`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set SSL usage -user <USER-ID> -password <PASSWORD> -value <VALUE>`

  The input parameters are:

Parameter	Description
`USER-ID`	(Optional). The URL of the CentraSite registry. For example, `http://localhost:53307/CentraSite/CentraSite`.
`PASSWORD`	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, `Administrator`.
`VALUE`	The values of either HTTP or HTTPS to allow HTTP or HTTPS communication respectively between AST and RR.

  Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set SSL usage -user Administrator -password manage -value http
```

  The response to this command could be:

```
Executing the command : set SSL usage
Successfully executed the command : set SSL usage
```

Securing Communication Between Software AG Runtime and External Clients

In the CentraSite environment, Software AG Runtime can receive requests from clients such as:

- User applications using an API to communicate with the registry or repository.
- Components of the Software AG Designer.

By default, only basic communication encryption without authentication is configured.

For information on how to configure SSL-based authentication and protect Tomcat, see Tomcat 7.0 documentation and product information at http://tomcat.apache.org/.

Software AG Runtime Properties File for SSL Communication

The file `com.softwareag.catalina.connector.https.pid-CentraSite.properties` located in the directory `Software AG_directory/profiles/CTP/configuration/com.softwareag.platform.config.propsloader` contains the properties that you need to set in order to configure Software AG Runtime for secure communication with external clients. The properties in this file define the SSL keystore and SSL truststore that Software AG Runtime uses.

Runtime properties for SSL communication in the *Software AG Infrastructure Administrator's Guide*. HTTPS connectors to set up the SSL environment. Note that this cross-product document refers to the properties file generically as `com.softwareag.catalina.connector.https.pid-<port_number>.properties`.

SSL Keystore

CentraSite comes with a sample keystore that contains self-signed certificates, which are located in `Software AG_directory/profiles/CTP/configuration/tomcat/conf`. The sample self-signed certificates are specific to localhost and therefore cannot be used for configuring SSL communication with CentraSite.

Acquire and provide your own server certificate and define its location with the parameter `keystoreFile` (replace the default value) in the Software AG Runtime properties file for SSL communication.

Note:
The CN of the certificate needs to be identical to the URL the server is addressed under, without the `https://`. For example, for a server reachable under `https://MyWebServer:8443/`, the CN needs to be `MyWebServer`. Software AG Runtime supports both Java keystores (keystoreType="JKS", which is the default) and PKCS#12 keystores (keystoreType="PKCS12"). Set the keystore password accordingly (parameter `keystorePass` in the Software AG Runtime properties file).

SSL Truststore

If you want to use client authentication for 2-way SSL, you need to set clientAuth="true" in the Software AG Runtime properties file for SSL communication and supply a truststore, which is a

keystore containing the certificate chain and trust root for the client certificates for which you want to allow access.

In the properties file, you also need to provide the following properties:

- truststoreFile: the name and path of the truststore file
- truststorePass: the password for accessing the truststore
- truststoreType: the type of the truststore
- truststoreProvider: the provider of the truststore

Note on SSL Port Number

If a URL addresses a location using SSL, the URL must explicitly specify the port number of the location, even if the default port number for SSL (443) is to be used.

CentraSite comes with a sample keystore that contains self-signed certificates which are located in `Software AG_directory/profiles/CTP/configuration/tomcat/conf` and need be replaced if SSL-based authentication is to be used.

Acquire and provide your own server certificate and define its location with the parameter `keystoreFile` (replace the default value) in the Software AG Runtime properties file for SSL communication.

Note:
The CN of the certificate needs to be identical to the URL the server is addressed under, without the https://. For example, for a server reachable under `https://MyWebServer:8443/`, the CN needs to be `MyWebServer`. Software AG Runtime supports both Java keystores (keystoreType="JKS", which is the default) and PKCS#12 keystores (keystoreType="PKCS12"). Set the keystore password accordingly (parameter `keystorePass` in the Software AG Runtime properties file).

Securing Communications with CentraSite for Synchronous Deployment

Configuring CentraSite to use SSL authentication provides secure communications for the deployment.

An administrator can configure CentraSite to use either of the following kinds of authentication:

- HTTP Basic authentication
- HTTPS (Secure Sockets Layer (SSL)) authentication

This section explains how SSL works with CentraSite (which acts as the client) and Mediator (which acts as the server). This section also provides the information you need to configure both the client and server sides for SSL authentication.

Anatomy of a SSL Connection

It is useful to conceptualize a CentraSite SSL connection in terms of a SSL client and a SSL server. The request for an SSL connection originates from a client.

During the SSL handshake process, the Mediator acting as the SSL server responds to the request for a connection by presenting its SSL credentials (an X.509 certificate) to the requesting CentraSite client. If those credentials are authenticated by the CentraSite client, either:

- An SSL connection is established and information can be exchanged between the CentraSite and Mediator.

  —OR—

- The next phase of the authentication process occurs, and the Mediator requests the SSL credentials of the CentraSite. If the Mediator verifies those credentials (that is, the client's identity), an SSL connection is established and information exchange can take place.

SSL Connection Type

The types of SSL connection referred to above are termed *one-way* and *two-way* SSL authentication:

- In a *one-way* SSL connection, client authenticates the credentials of server in preparation for setting up a secure transaction. In most cases, the server knows nothing about the client's identity because verification of its credentials is not required. When desired, however the client can be authenticated by means such as basic username or password.

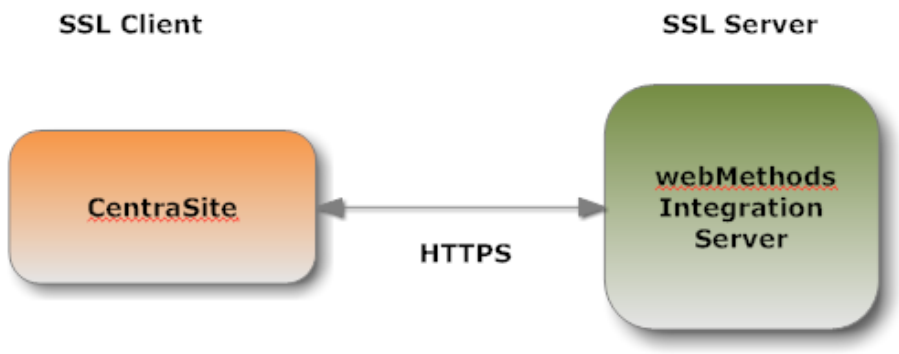
  This type of connection is where CentraSite needs to verify the authenticity of Mediator without itself needing to be authenticated. As a result, configurations on the CentraSite side are not actually required for this one-way connection.

- In a *two-way* SSL connection, both client and server must authenticate each other's credentials before an SSL connection is established and information can be exchanged.

  Unlike a one-way SSL connection, both CentraSite and Mediator require access to each other's SSL certificates in order to authenticate each other, establish an SSL connection, and transmit information. Compared to a one-way connection, a two-way SSL connection provides a much higher level of security.

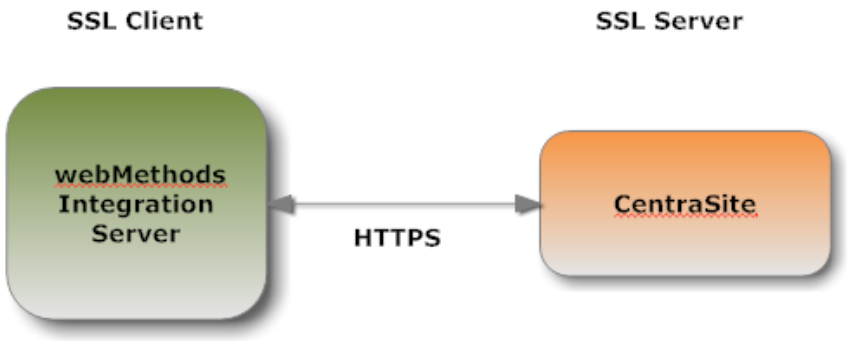
As an SSL Client

When CentraSite submits a HTTPS request to Mediator, CentraSite is the SSL client and Mediator with which it is communicating acts as the SSL server.



As an SSL Server

When Mediator submits a request to CentraSite through HTTPS, and a two-way SSL connection is established, the Mediator acts as the SSL client and the CentraSite acts as the SSL server.



Roadmap for Configuring SSL

The following table provides a high-level roadmap for configuring SSL on CentraSite.

Task	Activities	Notes
Create CentraSite keys and certificates	■ Generate a public key/private key pair. ■ Generate a certificate signing request (CSR) and send to the certificate authority (CA) for signing. ■ Receive validated certificate from the CA. ■ Import signed certificate into a keystore.	Required for one-way and two-way SSL connections. Refer to the documentation for Java *keytool* or your certificate management tool.
Create keystore and truststore for CentraSite	■ Create a keystore and import the signed certificate and private key.	Required for one-way and two-way SSL connections.

Task	Activities	Notes
	■ Create a truststore and import the certificate of the signing CA. ■ Store the keystore and truststore in a secure CentraSite certificates directory. **Important:** If you use a Java *keytool* to create the keystore, you cannot import an existing private key. You can use other tools such as OpenSSL or Portecle.	Refer to the documentation for your certificate management tool.
Obtain certificates of webMethods Mediator	Use the CentraSite truststore to save: ■ Signed certificate of the Mediator. ■ Signed certificate of the CA for the Mediator's SSL certificate.	Required for one-way and two-way SSL connections.

Creating Keys and Certificates

Use a standard certificate management tool, such as OpenSSL or Portecle, to generate a private/public key pair for CentraSite. Then, place the public key in a certificate signing request (CSR).

After creating the CSR, send to the CA to sign the CSR. Request the certificate in DER format. If you receive a certificate in PEM format (or any format other than DER), you need to convert it to DER format.

The signing CA's certificate attests to the identity of the CA that signed the digital certificate for the CentraSite. The CA should send this certificate to you when it sends you the digital certificate for the CentraSite.

Once you receive your signed certificate from the CA, you need to import the certificate into a keystore. You will then have an SSL certificate and private key to use with CentraSite.

Note:
The above process is described in general terms. The procedures may vary somewhat, depending upon the CA that you use. **Creating a Keystore and Truststore**

Keystores and truststores are files that function as repositories for storage of keys and certificates necessary for SSL authentication, encryption/decryption, and digital signing/verification services. Keystores and truststores provide added layers of security and ease of administration, compared to maintaining the keys and certificates in separate 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.

Obtaining the Certificates and Keys of the webMethods Mediator

If your CentraSite will submit HTTPS requests to the Mediator, the CentraSite will be acting as a client and will receive certificates from this Mediator. In order for these transactions to work, your CentraSite must have copies of their public keys and signing CA certificates. For information on importing Mediator certificates to CentraSite and setting up client authentication, see *webMethods Integration Server Administrator's Guide*

Keystores and Truststores

CentraSite stores its private keys and SSL certificates in keystore files and the trusted roots for the certificates in truststore files. Keystores and truststores are secure files with industry-standard file formats.

Keystore File

CentraSite uses a special file called a *keystore* to store SSL certificates and keys.

A keystore file contains one or more pairs of a private key and signed certificate for its corresponding public key. The keystore should be strongly protected with a password, and stored (either on the file system or elsewhere) so that it is accessible only to administrators.

Keystore File Formats

The default, certificate file format for a CentraSite keystore is. JKS (Java keystore). Java keystore is a commonly used, standardized, certificate file format that provides a high degree of portability. PKCS#12 is another format you can use for a keystore. Other keystore types can be made available by:

- Loading additional security providers
- Setting the watt.security.keyStore.supportedTypes property.

HSM-Based Keystores

Under certain conditions, Mediator supports the use of keystore files stored on a Hardware Security Module (HSM). Integration Server supports HSM-based keystores for ports, but not for other components. You cannot use HSM-based keystores with remote server aliases, outbound certificates, an internal server port, WS-Security, or Integration Server public services.

Creating a Keystore

You can create and manage CentraSite keystores at the command line using keytool, a Java certificate editor.

You can also use other standard tools such as OpenSSL and Portecle.

Note:
Software AG does not provide its own set of keystore utilities for creating or managing keystore files.

Truststore File

CentraSite uses a *truststore* to store its trusted root certificates, which are the public keys for the signing CAs. Although a truststore can contain the trusted roots for entire certificate chains, there

is no requirement for the organization of certificates within a CentraSite truststore. It simply functions as a database containing all the public keys for CAs within a specified trusted directory.

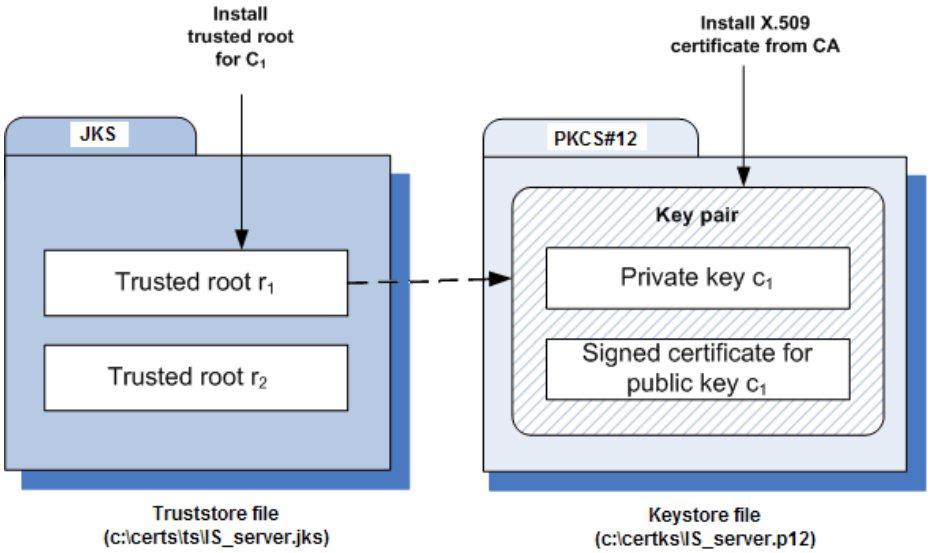
Truststore File Formats

CentraSite uses a *truststore* to store its trusted root certificates, which are the public keys for the signing CAs. Although a truststore can contain the trusted roots for entire certificate chains, there is no requirement for the organization of certificates within a CentraSite truststore. It simply functions as a database containing all the public keys for CAs within a specified trusted directory.

How CentraSite Uses a Keystore and Truststore

For a CentraSite service to be SSL authenticated, it must have a valid, authorized X.509 certificate installed in a keystore file *and* the private key and signing certificate for the certificate issuer (CA) installed in a truststore file. The following figure illustrates these requirements and the relationship between the two files.

Example Truststore File and Keystore File Showing Relationship



As illustrated in the figure, the same truststore file can contain multiple trusted root certificates (public keys for the signing CAs). These trusted roots might be associated with numerous keystore files. A keystore file can contain the key pairs for multiple CentraSite services, and the entire certificate chain required for a service's authentication.

With a certificate chain, it is necessary to validate each subsequent signature in the list until a trusted CA certificate is reached. For CentraSite, you must include the entire chain of certificates in a keystore and truststore. Also, any root CA certificates in use by clients must be in a CentraSite truststore.

Protecting Keystore and Truststore Files

Keystore and truststore files exist within the file system of your operating system, and since these are critically important files, you want to maintain them in a secure directory path. If either of the these files cannot be located, CentraSite authentication is disabled and no connection with CentraSite

can be made. It is recommended that only users serving as CentraSite administrators have access to these certificate files.

Configuring CentraSite to Use SSL

The configuration settings covered in this section deal with the CentraSite client side.

You can configure CentraSite client to use SSL in the following ways:

- One-way SSL
- Two-way SSL

Configuring CentraSite Client to Use One-way SSL

To configure CentraSite for one-way SSL authentication

1. Create at least one truststore `centrasitetruststore.jks`, in JKS format, in a desired location on the machine where CentraSite is running.

2. Import the Mediator's self-signed certificate `mediator.cer` into the above created truststore or JAVA cacerts.

   When prompted for password, the default for truststores is `password`.

```
C:\deploykeystores\new>keytool -export -alias mediator
 -keystore mediatorkeystore.jks -rfc -file mediator.cer
Enter keystore password:
Certificate stored in file <mediator.cer>

C:\deploykeystores\new>keytool -import -alias mediator
-keystore centrasitetruststore.jks -file mediator.cer
Enter keystore password:
Re-enter new password:
Owner:
Issuer:
Serial number:
Valid from:
Certificate fingerprints:
             Trust this certificate? [no]:  yes
Certificate was added to keystore

C:\deploykeystores\new>
```

   If opting to import certificate in to Java cacerts, the Java runtime needs to trust the certificates of the Mediator in order to establish the SSL connections. To do that, add the certificate to the trusted certificates of Java via the *keytool* utility that comes with Java. The following command adds the certificate located at a location (for example, c:\temp\server.crt) to the trusted certificates in the Java used by CentraSite:

```
keytool.exe -import -v -trustcacerts -alias test -file "C:\temp\server.crt"
-keystore  "<JDKInstallDir>\jre\lib\security\cacerts"
```

When prompted for password, the default for Java is `changeit`.

3. Add the following Java system properties to the `custom_wrapper.conf` file in `<SuiteInstallDir>/profiles/CTP/configuration` folder. For information about setting Java system properties, see the webMethods cross-product document, *Software AG Infrastructure Administrator's Guide*.

```
wrapper.java.additional.<n>=-Djavax.net.ssl.trustStore=
      <location_of_truststore>
wrapper.java.additional.<n>= -Djavax.net.ssl.trustStorePassword=
      <password_for_truststore>
```

In the settings above:

- *<n>* is a unique sequence number that you assign to each wrapper.java.additional property. For more information about assigning this sequence number, see the wrapper.java.additional property description in the cross-product document, Working with the webMethods Product Suite and the Java Service Wrapper.
- *<location_of_truststore>* is the location to the trust store file (for example, `C:/deploykeystores/new/centrasitetruststore.jks`).
- `<password_for_truststore>` is the password for the trust store.

4. Go to the section #Java Additional Parameters. Add the following property lines:

```
wrapper.java.additional.7=-Djavax.net.ssl.

trustStore="C:/deploykeystores/new/centrasitetruststore.jks"
wrapper.java.additional.8=-Djavax.net.ssl.trustStorePassword=password
```

5. Set the values as needed:

`wrapper.java.additional.7=-Djavax.net.ssl.trustStore=` represents the location of a truststore file (for example, `centrasitetruststore.jks`).

`wrapper.java.additional.8=-Djavax.net.ssl.trustStorePassword=` represents the password for a truststore.

6. Save and close the file.

7. Now restart the CentraSite Tomcat. All communication through the Mediator to the database should now be using SSL.

Configuring CentraSite Client to Use Two-way SSL

To configure CentraSite for two-way SSL authentication

1. Using OpenSSL, create a self-signed certificate (`centrasite.cer`) with the following command:

```
openssl req -new -x509 -days 2000 -sha1 -newkey rsa:1024 -nodes
```

```
-keyout server.key -out server.crt -subj "/O=Company/OU=Unit/CN=localhost"
```

Whatever is specified in the CN section of the subject must match the hostname of the machine running the Mediator and is used to send requests to the Mediator.

2. Create at least one keystore `centrasitekeystore.jks`, in PKCS#12 or JKS format, containing a CentraSite key pair to use for SSL.

```
C:\deploykeystores\new>keytool -v -genkeypair -alias centrasite
              -keyalg RSA -validity 1000 -keystore centrasitekeystore.jks
Enter keystore password:
Re-enter new password:
What is your first and last name?
What is the name of your organizational unit?
What is the name of your organization?
What is the name of your City or Locality?
What is the name of your State or Province?
What is the two-letter country code for this unit?

Enter key password for <centrasite>
            <RETURN if same as keystore password>:
[Storing centrasitekeystore.jks]

C:\deploykeystores\new>
```

3. Create at least one truststore `centrasitetruststore.jks`, in JKS format, in a desired location on the machine where CentraSite is running.

4. Import the Mediator's self-signed certificate `mediator.cer` into the above created truststore or Java cacerts.

When prompted for password, the default for truststores is `password`.

```
C:\deploykeystores\new>keytool -export -alias mediator
   -keystore mediatorkeystore.jks -rfc -file mediator.cer
Enter keystore password:
Certificate stored in file <mediator.cer>

C:\deploykeystores\new>keytool -import -alias mediator
   -keystore centrasitetruststore.jks -file mediator.cer
Enter keystore password:
Re-enter new password:
Owner:
Issuer:
Serial number:
Valid from:
Certificate fingerprints:
             Trust this certificate? [no]:  yes
Certificate was added to keystore

C:\deploykeystores\new>
```

If opting to import certificate in to Java cacerts, the Java runtime needs to trust the certificates of the Mediator (regardless of whether this is a Tomcat application or a standalone application) in order to establish the SSL connections. To do that, add the certificate to the trusted certificates of Java through the *keytool* utility that comes with Java. The following command adds the

certificate located at a location (for example, c:\temp\server.crt) to the trusted certificates in the Java used by CentraSite:

```
keytool.exe -import -v -trustcacerts -alias test
-file "C:\temp\server.crt"
-keystore "<JDKInstallDir>\jre\lib\security\cacerts"
```

When prompted for password, the default for Java is `changeit`.

5. Export the CentraSite's self-signed certificate `centrasite.cer` in to the Mediator's truststore.

6. Open the `wrapper.conf` file located in the directory `<CentraSiteInstall_Directory>/profiles/CTP/configuration`

7. Go to the section #Java Additional Parameters. Add the following property lines:

```
wrapper.java.additional.5=-Djavax.net.ssl.keyStore="C:/deploykeystores/new/
centrasitekeystore.jks"
wrapper.java.additional.6=-Djavax.net.ssl.keyStorePassword=password
wrapper.java.additional.7=-Djavax.net.ssl.trustStore="C:/deploykeystores/ne
w/centrasitetruststore.jks"
wrapper.java.additional.8=-Djavax.net.ssl.trustStorePassword=password
```

8. Set the values as needed:

   `wrapper.java.additional.5=-Djavax.net.ssl.keyStore=` represents the location of a keystore file (say, `centrasitekeystore.jks`).

   `wrapper.java.additional.6=-Djavax.net.ssl.keyStorePassword=` represents the password for a keystore.

   `wrapper.java.additional.7=-Djavax.net.ssl.trustStore=` represents the location of a truststore file (say, `centrasitetruststore.jks`).

   `wrapper.java.additional.8=-Djavax.net.ssl.trustStorePassword=` represents the password for a truststore.

9. Save and close the file.

10. Now restart the CentraSite Tomcat. All communication through the Mediator to the database should now be using SSL.

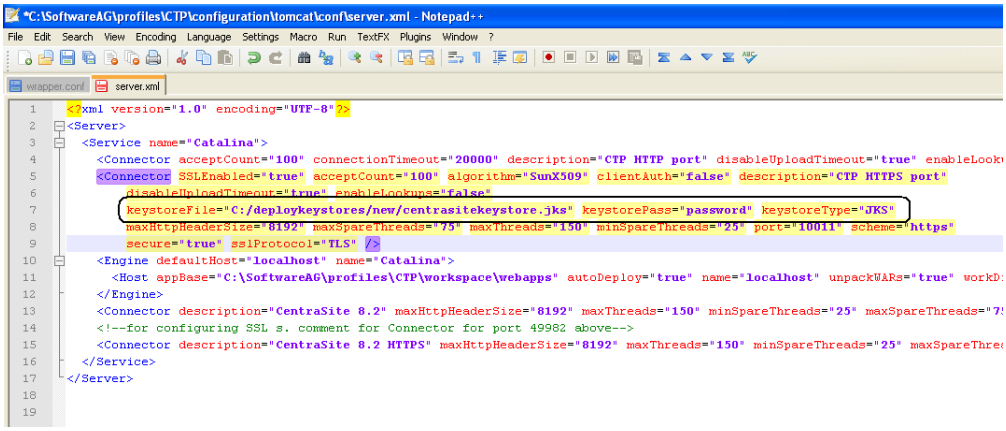
Configuring CentraSite Client for SSL Using CTP Server.xml File

To configure CentraSite for SSL authentication using CTP server.xml file

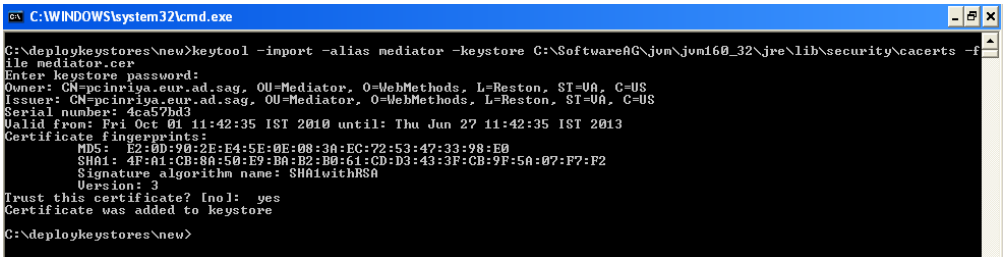
1. Open the `server.xml` file.

   You can find the `server.xml` file in the `<CentraSiteInstall_Directory>/profiles/CTP/configuration/tomcat/conf` directory.

2. Type the keystore information as specified:



3. Add the Mediator certificate (`mediator.cer`) into CentraSite JVM cacerts as below:



Configuring webMethods Integration Server to Use SSL

The configuration settings covered in this section deal with the webMethods Integration Server side.

You can configure Integration Server to use SSL in the following ways:

- One-way SSL
- Two-way SSL

Configuring Integration Server to Use One-way SSL

To configure Integration Server for one-way SSL authentication

1. Using OpenSSL, create a self-signed certificate (`mediator.cer`) with the following command:

```
openssl req -new -x509 -days 2000 -sha1 -newkey rsa:1024 -nodes
-keyout server.key -out server.crt -subj "/O=Company/OU=Unit/CN=localhost"
```

   Whatever is specified in the CN section of the subject must match the hostname of the machine running the Mediator and is used to send requests to Mediator.

2. Create at least one keystore `mediatorkeystore.jks`, in PKCS#12 or JKS format, containing an Integration Server key pair to use for SSL and its corresponding key alias.

```
C:\deploykeystores\new>keytool -v -genkeypair -alias mediator
-keyalg RSA -validity 1000 -keystore mediatorkeystore.jks
Enter keystore password:
Re-enter new password:
What is your first and last name?
What is the name of your organizational unit?
What is the name of your organization?
What is the name of your City or Locality?
What is the name of your State or Province?
What is the two-letter country code for this unit?

Enter key password for <mediator>
            <RETURN if same as keystore password>:
[Storing mediatorkeystore.jks]

C:\deploykeystores\new>
```

3. Export the Mediator's self-signed certificate `mediator.cer` into the CentraSite's truststore.

4. Configure an HTTPS port and specify the client authentication to **Username/Password**. The server prompts the client for a user ID and password.

5. On the **Ports** screen, click **Edit** to change the **Access Mode**. You may Set Access Mode to **Allow by Default** or **Reset to default access settings**.

   For more information on configuring ports and client authentication, see *webMethods Integration Server Administrator's Guide*.

6. Restart Integration Server.

Configuring Integration Server to Use Two-way SSL

To configure Integration Server for two-way SSL authentication

1. Using OpenSSL, create a self-signed certificate (`mediator.cer`) with the following command:

```
openssl req -new -x509 -days 2000 -sha1 -newkey rsa:1024 -nodes
-keyout server.key -out server.crt -subj "/O=Company/OU=Unit/CN=localhost"
```

   Whatever is specified in the CN section of the subject must match the hostname of the machine running the Mediator and is used to send requests to the Mediator.

2. Create at least one keystore `mediatorkeystore.jks`, in PKCS#12 or JKS format, containing an Integration Server key pair to use for SSL.

```
C:\deploykeystores\new>keytool -v -genkeypair -alias mediator
-keyalg RSA -validity 1000 -keystore mediatorkeystore.jks
Enter keystore password:
Re-enter new password:
```

```
What is your first and last name?
What is the name of your organizational unit?
What is the name of your organization?
What is the name of your City or Locality?
What is the name of your State or Province?
What is the two-letter country code for this unit?

Enter key password for <mediator>
             <RETURN if same as keystore password>:
[Storing mediatorkeystore.jks]

C:\deploykeystores\new>
```

3. Create at least one truststore `mediatortruststore.jks`, in the JKS format, in a desired location on the machine where CentraSite is running.

4. Export Mediator's self-signed certificate `mediator.cer` into the CentraSite's truststore.

5. Import CentraSite's self-signed certificate `centrasite.cer` in to the Mediator's truststore `mediatortruststore.jks`.

```
C:\deploykeystores\new>keytool -export -alias

centrasite -keystore centrasitekeystore.jks -rfc -file

centrasite.cer
Enter keystore password:
Certificate stored in file <centrasite.cer>

C:\deploykeystores\new>keytool -import -alias

mediator -keystore mediatortruststore.jks -file

centrasite.cer
Enter keystore password:
Re-enter new password:
Owner:
Issuer:
Serial number:
Valid from:
Certificate fingerprints:
              Trust this certificate? [no]:  yes
Certificate was added to keystore

C:\deploykeystores\new>
```

6. Create a keystore and truststore alias using the above created keystore (`mediatorkeystore.jks`) and truststore (`mediatortruststore.jks`) respectively. For more information on creating keystore and truststore aliases, see *webMethods Integration Server Administrator's Guide* in the documentation set for webMethods Integration Server.

7. Configure an HTTPS port and specify the client authentication to any of the following:

   - **Username/Password**. The server prompts the client for a user ID and password.

- **Request Client Certificates**. The server requests client certificates for all requests. If the client does not provide a certificate, the server prompts the client for a user ID and password. If the client provides a certificate:
  - 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.
  - 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.
- **Require Client Certificates**. The server requires client certificates for all requests. The server behaves as described for Request Client Certificates, except that the client must always provide a certificate.

8. On the **Ports** screen, click **Edit** to change the **Access Mode**. You may Set Access Mode to Allow by Default or Reset to default access settings.

9. If the selected client authentication as **Require Client Certificates** above, map the client certificate to any valid user in the Integration Server.

   For more information on configuring ports and client authentication, see *webMethods Integration Server Administrator's Guide*.

10. Restart the Integration Server.

Configuring webMethods Mediator to Use SSL

Configure your instance of webMethods Mediator as described in *Administering webMethods Mediator*.

Configuring Registry Cache Settings

For performance reasons, CentraSite uses an internal cache to store registry objects accessed through JAXR. In some cases, you might want to modify the cache configuration settings, in order to determine the optimal setup for your system.

CentraSite provides a set of command tools to display and to modify the JAXR-based configuration settings. The JAXR-based configuration settings apply to each connection.

Displaying Registry Cache Configuration

Pre-requisites:

To display the cache configuration of CentraSite registry through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

The CentraSite Registry or Repository (CRR) must be online.

CentraSite provides a Java tool named `CentraSiteCacheConfiguration.jar` for this purpose. You must execute the Java tool `CentraSiteCacheConfiguration.jar` with a `DISPLAY` keyword.

To display the cache settings for JAXR

- Run the Java tool `CentraSiteCacheConfiguration.jar`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd CentraSiteCacheConfiguration.jar <CentraSite URL> <admin user id> <password> DISPLAY`

  The input parameters are:

Parameter	Description
`CentraSite URL`	The URL of the CentraSite registry. For example, `http://localhost:53307/CentraSite/CentraSite`.
`admin user id`	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, `Administrator`.
`password`	The password for the CentraSite user identified by the parameter `admin user id`.

  Example (all in one line)

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd
CentraSiteCacheConfiguration.jar "http://localhost:53307/CentraSite/CentraSite"
DOMAIN\admin pAsSw0rD DISPLAY
```

Modifying Registry Cache Configuration

Pre-requisites:

To modify the cache configuration of CentraSite registry through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

The CentraSite Registry or Repository (CRR) must be online.

CentraSite provides a Java tool named `CentraSiteCacheConfiguration.jar` for this purpose. You must execute the Java tool `CentraSiteCacheConfiguration.jar` with a `SET` keyword.

The SET keyword is followed by pairs of option and value. After the modification operation is completed, the tool displays the modified cache configuration.

The following options may be specified:

- **Option: maxElementsOnHeap**: The option `maxElementsOnHeap` defines the maximum number of elements in the cache. A value of 0 means no limit. Once the cache is full, an element is evicted according to the algorithm specified by the `memoryStoreEvictionPolicy` option.

- **Option: maxElementsOffHeap**: The option `maxElementsOffHeap` defines the amount of off-heap memory available to the cache. Off-heap memory is a separate unit of memory available outside of the conventional JVM heap which can be used for caching.

  This option's values are given as `<number>k|K|m|M|g|G|t|T`, where the units can be kilobytes (k|K), megabytes (m|M), gigabytes (g|G), or terabytes (t|T).

  For example, `maxMemoryOffHeap="2g"` allots 2 gigabytes to off-heap memory. A value of 0 means no off-heap memory.

  Before using off-heap memory, direct memory space, also called direct (memory) buffers, must be allocated. In most popular JVMs, direct memory space is allocated using the Java property `-XX:MaxDirectMemorySize`. This memory space, which is part of the Java process heap, is separate from the object heap allocated by `-Xmx`. The value allocated by `-XX:MaxDirectMemorySize` must not exceed the physical RAM and is likely to be less than the total available RAM due to other memory requirements. The value allocated to direct memory should be at least 32MB more than the off-heap memory allocated to the caches.

  Note:
  The use of this option requires a special license key. For further information contact your software supplier.

- **Option: memoryStoreEvictionPolicy**: The option `memoryStoreEvictionPolicy` defines the algorithm to be used in case an element needs to be evicted from the cache. Possible values are:
  - LRU - least recently used
  - LFU - least frequently used
  - FIFO - first in first out
- **Option: statistics**: The option `statistics` defines whether the cache should capture statistical information and if yes at which accuracy level. The statistics comprise information like cache hits, cache misses and average time to get an element. Possible values are:
  - OFF - no statistics
  - ACCURACY_NONE - fast but not accurate
  - ACCURACY_BEST_EFFORT - best effort accuracy
  - ACCURACY_GUARANTEED - guaranteed accuracy

Note:
If none of the options is specified in the SET operation, the existing value is copied.

To modify the cache settings for JAXR

- Run the Java tool `CentraSiteCacheConfiguration.jar`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd CentraSiteCacheConfiguration.jar <CentraSite URL> <admin user id> <password> SET <option> <value>[<option> <value> ...]`

The input parameters are:

Parameter	Description
`CentraSite URL`	The URL of the CentraSite registry. For example, `http://localhost:53307/CentraSite/CentraSite`.
`admin user id`	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, `Administrator`.
`password`	The password for the CentraSite user identified by the parameter `admin user id`.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd
CentraSiteCacheConfiguration.jar http://localhost:53307/CentraSite/CentraSite
DOMAIN\admin pAsSw0rD SET maxElementsOnHeap 10000
```

Configuring User Authentication and Repositories

User Authentication

Authentication is the process of validating a user's login credentials (for example, the user's certificate, or user ID and password) that match the credentials known to the system. CentraSite can use a number of different data sources, known as domains, to validate a user's credentials; these currently include the following:

- An *internal* text file
- Microsoft Active Directory (AD), when used through LDAP
- LDAP

This document is intended for customers who wish to configure CentraSite's user authentication features.

Assumptions

If an external repository, for example LDAP, is used, this topic assumes that it has already been set up and that you have the necessary expertise and privileges to perform administrative tasks. Usually, the use of CentraSite does not influence any design decisions that were made in setting up an external user repository; CentraSite just needs to know how to access the users and groups of the user repository.

User Repositories

A user repository is in general terms a set of user credentials (optionally including user certificates and so on), with the possible addition of information such as the groups to which a user belongs, the user's address, telephone number and the email address. Often, an enterprise implements a central user repository that can be used by applications throughout a network to authenticate users; when a user tries to log in to an application, the application issues a request to the user repository to check whether the user credentials supplied are valid. Usually the user repository is created and maintained separately from the applications that use it.

A newly-installed CentraSite system is configured to authenticate users against an internal text file. This is intended to enable an administrator to log in and modify the configuration as required to meet enterprise requirements; typically, and in particular if you are working in a distributed environment, where one or more Application Server Tiers and a separate Registry or Repository are involved, an external repository such as Active Directory or LDAP will form the core of the authentication process.

Selecting a User Repository for Authentication

Access to information stored in CentraSite generally requires a user name and password, to ensure that data can only be stored, modified or retrieved by authorized users. CentraSite supports the following types of user repository:

- An internal text file
- LDAP (for example Sun, OpenLDAP, ADS)

CentraSite maintains information about each kind of user repository in so-called *authentication configurations*. An authentication configuration specifies the type of user repository to be used and any parameters that are required to configure the user repository. CentraSite is delivered with one predefined authentication configuration, namely the configuration to use an internal text file and this configuration is the default configuration. You can define additional authentication configurations; also, you can set any one of the defined configurations to be the default configuration.

In general, user authentication information is stored in the user repository, not in CentraSite. CentraSite can contain a copy of selected data fields from the user repository for each registered CentraSite user. The user information in the CentraSite user registry is stored in objects of the type `User`. You can associate a CentraSite user object with a user in a user repository. In this case you can map data fields from the user repository into the user object in the CentraSite registry. The data in the mapped data fields is visible when you display the user object in CentraSite.

Domain Names of User Repositories

Each user repository is uniquely identified by a domain name. A user in a user repository is uniquely identified by the combination of domain name and user name.

When you log in to CentraSite Control, you must supply the name of a domain in which you are registered and your user name, in the format <DomainName>\<UserName> , for example, `Headquarters\JSmith`.

The domain name for an authentication configuration of type Internal is always `INTERNAL`. Since this name is fixed, there can be only one such configuration defined per instance of the CentraSite registry.

Default User Repository

While CentraSite is running, there is always exactly one default user repository. When you install CentraSite, the default user repository is set to the internal text file. You can change the default to any other user repository for which an authentication configuration exists.

Users who are registered in the default user repository can omit the domain name when they log in. For example, if the domain `Headquarters` is the default domain and it contains a user whose user name is `JSmith`, then this user can log in as `JSmith` instead of `Headquarters\JSmith`. Users who are not registered in the default user repository must always use the format <DomainName>\<UserName> to log in.

Notes on User Authentication in CentraSite

Case Sensitivity

User names and domain names are treated as either case sensitive or case insensitive, according to the configured authentication mechanism.

INTERNAL authentication	case sensitive
Active Directory authentication	case insensitive
LDAP authentication	case insensitive

Working in an Offline Environment

If you wish to work in an offline environment, for example on a laptop computer that is not connected to the network, you should be aware of certain restrictions that apply in the area of authentication.

Important:
When CentraSite is installed in an environment where the users are authenticated against a central service, for example an LDAP server, authentication does not work if the machine is disconnected from the network. So if you intend to use CentraSite on a mobile device when it is not connected to the network, ensure that at least one user is available who can also be authenticated offline, for example from an internal or local LDAP user repository.

User Authentication Configurations

When CentraSite utilizes a user repository, certain connection parameters are required. The connection parameters are stored in an authentication configuration. If you need to work with more than one user repository (for example, a user repository for test purposes and a user repository for a production environment), you can define several authentication configurations.

At any given time, only one authentication configuration can be the default authentication configuration.

After creating or modifying the authentication settings, the new settings apply immediately to the CentraSite Registry or Repository.

Creating and Maintaining Authentication Configurations

Pre-requisites:

To configure the user authentication settings through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

The authentication in the CentraSite Registry or Repository is configured with default settings during installation. You can define additional authentication configurations, and you can change the default configuration to be one of the additional configurations.

The default authentication configuration determines the user repository that is used to authenticate users who log on to CentraSite. Initially, the default user repository is CentraSite's own user repository, which has the domain name INTERNAL. You might want to define additional configurations that define for example an LDAP user repository.

CentraSite provides a set of command tools for this purpose.

You can use these tools to perform the following tasks:

- Create an authentication configuration
- Modify an authentication configuration
- Delete an authentication configuration

  Keep the following points in mind:

  - If you do not require a particular authentication configuration any more, you can delete it from the list of available configurations.
  - You cannot remove the pre-installed domain `INTERNAL`.
  - If you remove a configuration that is the current default configuration, the configuration is removed and the default reverts to the INTERNAL configuration.
  - To delete an existing authentication configuration, use the command named `remove Authentication`.

  Note:
  When you delete an authentication configuration, CentraSite does not delete the user objects that are associated with this configuration. Thus, these users are displayed in the list of users in CentraSite Control, even though the domain to which they belong is no longer accessible to CentraSite.

- Set a default authentication configuration

  Keep the following points in mind:

- If you have defined more than one authentication configuration, you can change the current default configuration to one of the other configurations.
- The user domain of the new default configuration must include at least one user who is defined in CentraSite with the CentraSite Administrator role, otherwise you are prompted to enter a user who is defined as administrator in that configuration.
- To set a new default authentication configuration, use the command named `set DefaultDomain`.
- If the user domain of the configuration that you wish to set to the default does not contain any user who is defined in CentraSite with the CentraSite Administrator role, a dialog appears, asking you to provide the user name and password of a domain user who has granted this role in CentraSite.
- If the user already exists in CentraSite, but does not have the CentraSite Administrator role, the role is granted to the user. If the user does not exist in CentraSite, a user with the given user name is created in CentraSite and is granted the CentraSite Administrator role.
- The dialog also allows you to specify an organization for the user, in cases where the user did not already exist in CentraSite. The newly created CentraSite user is assigned to this organization. If you do not specify an organization, the user is assigned to the default organization.
- Users who are in the default domain can log in without having to specify the domain name, but they can specify the domain name if they wish. Users who are not in the current default domain always have to specify the domain name when logging in.
    - If your default authentication configuration contains only one user who has the CentraSite Administrator role in CentraSite, it is not possible to delete this user from CentraSite, or to remove the CentraSite Administrator role from the user. This is because the default configuration must always contain at least one user who is defined in CentraSite with the CentraSite Administrator role.
    - If you try to log in to a CentraSite component (for example, CentraSite Control) by supplying a user name and password but no domain name, the authentication mechanism assumes that you belong to the domain of the default configuration and authenticates you against this domain. If you change the default configuration as described above and subsequently try to log in to a CentraSite component, you must supply your domain name in addition to your user name, so that the authentication mechanism knows which domain to use to check your credentials.
- When you set a new default authentication configuration, you might want to change the association between CentraSite users (that is, CentraSite registry objects representing users) and users in the external user repository.

- List the names of all defined authentication configurations
- List details of a specific authentication configurations
- Validate that an authentication configuration is correctly specified

- Listing Names of Existing Authentication Configurations: Run the command `list Authentication`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Authentication`

  Note:
  The list also indicates the default configuration.

  The response to this command could be:

```
Executing the command : list Authentication
Successfully executed the command : list Authentication
```

- Obtaining Details of an Authentication Configuration: To fetch the details of an existing authentication configuration, run the command `get Authentication`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get Authentication -domain <DOMAIN>`

  The input parameters are:

Parameter	Description
`DOMAIN`	The domain name of the user repository associated with the configuration.

  Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get Authentication -domain LDAPDomain
```

  The response to this command could be:

```
Executing the command : get Authentication
Domain Name         Domain Type
-------------------------------------
LDAPDomain            LDAP
Properties:
   useaf: "false"
   userrootdn: "ou=people,ou=gdm,o=sag"
   personobjclass: "inetOrgPerson"
   uidprop: "cn"
   url: "ldap://daeqarh01:10389"
   noPrinIsAnonymous: "false"
   groupobjclass: "groupOfUniqueNames"
   usecaching: "false"
   applyDomain: "true"
   gidprop: "cn"
   createGroupProperties: "true"
   alias: "LDAPDomain"
   memberinfoingroups: "true"
   creategroups: "true"
   createUserProperties: "true"
   mattr: "uniqueMember"
   grouprootdn: "ou=groups,ou=gdm,o=sag"
User Mappings:
```

```
   displayName: "personName:fullName"
   mail: "emailAddresses:emailAddress:address"
   sn: "personName:lastName"
Group Mappings:
   description: "description"
Successfully executed the command : get Authentication
```

- Setting a Default Authentication Configuration: To set the default authentication configuration in CentraSite, run the command `set DefaultDomain`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set DefaultDomain -domain <DOMAIN>`

  The input parameters are:

Parameter	Description
`DOMAIN`	The domain name of the user repository associated with the configuration.

  Important:
  An authentication configuration containing the specified domain must already exist in CentraSite.

  Note:
  If you have set up multiple CentraSite instances in cluster mode, ensure that you execute the `set DefaultDomain` command individually in each of these CentraSite instances in the cluster.

  Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set DefaultDomain -domain
LDAPdomain
```

  The response to this command could be:

```
Executing the command : set DefaultDomain
Successfully executed the command : set DefaultDomain
```

- Adding an Authentication Configuration: To add a new authentication configuration to CentraSite, run the command `set Authentication`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Authentication -domain <DOMAIN>`

  The input parameters are:

Parameter	Description
`DOMAIN`	The domain name of the user repository associated with the configuration.

When adding a LDAP configuration, the values you entered for the command parameters are evaluated against the specified LDAP server. Make sure that the corresponding LDAP server is available and running.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Authentication -domain
LDAPdomain
```

The response to this command could be:

```
Executing the command : set Authentication
Successfully executed the command : set Authentication
```

- Modifying an Authentication Configuration: To modify an existing authentication configuration, run the command `set Authentication`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Authentication -domain <DOMAIN>`

  The input parameters are:

Parameter	Description
`DOMAIN`	The domain name of the user repository associated with the configuration.

  When modifying a LDAP configuration, the values you entered for the command parameters are evaluated against the specified LDAP server. Ensure that the corresponding LDAP server is available and running.

  Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Authentication -domain
LDAPdomain
```

  The response to this command could be:

```
Executing the command : set Authentication
Successfully executed the command : set Authentication
```

- Removing an Authentication Configuration: To remove an existing authentication configuration, run the command `remove Authentication`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd remove Authentication -domain <DOMAIN>`

  The input parameters are:

Parameter	Description
`DOMAIN`	The domain name of the user repository associated with the configuration.

Note:
Keep the following points in mind:

- You cannot remove the pre-installed domain `INTERNAL`.
- You also cannot remove a configuration that is the current default configuration. If you want to delete such a configuration, you must first change the default configuration to another configuration.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd remove Authentication -domain
LDAPdomain
```

The response to this command could be:

```
Executing the command : remove Authentication
Successfully executed the command : remove Authentication
```

Specifying Domain Name

Unless the type of authentication that you wish to specify is `INTERNAL`, the authentication configuration requires a domain name. This is the domain name that is used to address the users who are authenticated against the specified user repository.

Important:
When working with LDAP, the domain name should be the name of a specific domain controller (DC) node in the LDAP tree structure. There can be many DC nodes in an LDAP tree structure, and you must select the DC node that is the deepest ancestor node (parent, grandparent, and so on) of all of the user nodes. Here, deepest means furthest away from the LDAP tree's root node. For example, if the usernames in an LDAP tree structure are located in the LDAP path `uid=Username,ou=People,dc=mydomain,dc=com`, then both `dc=mydomain` and `dc=com` are ancestor DC nodes of the user nodes, but since `dc=mydomain` is deeper than `dc=com`, you should specify the domain name as `MYDOMAIN` and not `COM`. If the path to the user nodes does not include any DC nodes, specify the root node. For example, if a user's full path is `cn=Username,ou=People,ou=RnD,o=Company`, set the domain name to `Company`.

The domain name for an authentication configuration of type Internal is always `INTERNAL`. Since this name is fixed, there can be only one such configuration defined per instance of the CentraSite registry.

Mapping User and Group Fields

When you specify an authentication configuration, you specify the correlation between properties stored in the CentraSite JAXR-based model for the object type `User` and properties stored in the external user repository.

The JAXR-based properties stored in CentraSite for the object type `User` are organized according to the following structure:

```
description
organization
personName
```

```
 firstName
 middleName
 lastName
 fullName
postalAddresses
 postalAddress
     street
     streetNumber
     postalCode
     city
     stateOrProvince
     country
     postalScheme
emailAddresses
 emailAddress
     address
telephoneNumbers
 telephoneNumber
     countryCode
     areaCode
     number
     extension
     url
URL
```

The mappings are used in CentraSite Control when you create a new CentraSite user and wish to associate the user with a user in the external user repository and also when you click on **Synchronize** for a user in the CentraSite Control. The corresponding dialog in CentraSite Control for locating the external user definition includes a search capability in which you can specify the JAXR-based mapping properties mentioned above to locate a particular user. The search mechanism translates the JAXR-based property searches into corresponding searches of the properties of the external user repository, using the mappings you define here.

Specify the mappings as required. Typically, you only specify mappings for properties that you wish to make available for searches of the external user repository. If you do not require the capability of searching the external user repository, you can leave all of the fields empty.

Note:
User property mappings are not available for users who are stored in the internal repository.

File Structure of Login Authentication Configuration

Beginning with version 9.10, most of the properties for login authentication configuration are moved from the existing jaas.config file to the new Dynamic Configuration Property file. The file structure of the login authentication configuration is as follows:

jaas.config File

```
<Software AG_directory>\profiles\CTP\configuration\jaas.config
```

Dynamic Configuration Property File

```
<Software AG_directory>\profiles\CTP\configuration\
com.softwareag.platform.config.propsloader\com.softwareag.security.ldap.server.pid-
<domain_id>.properties
```

User Mapping Property File

`<Software AG_directory>\profiles\CTP\configuration\com.softwareag.platform.config.propsloader\com.softwareag.security.ldap.server.<domain_id>-user.properties`

Group Mapping Property File

`<Software AG_directory>\profiles\CTP\configuration\com.softwareag.platform.config.propsloader\com.softwareag.security.ldap.server.<domain_id>-group.properties`

Configuring Internal Authentication Type

Pre-requisites:

To perform administration tasks on the internal authentication file through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

The Internal Authentication type allows you to authenticate a user against a set of user names and passwords that are maintained in a text file on the CentraSite Registry or Repository. Passwords are stored in SHA-512 hashed format, they cannot be decrypted. All user names and passwords are case-sensitive.

A typical use of such an authentication type would be during the initial set-up and testing of all required CentraSite components. In a production environment, one would typically use a central repository, for example, Microsoft Active Directory or LDAP, instead of Internal authentication.

The domain name for the Internal authentication type is always `INTERNAL`; this cannot be changed. A user who is registered in the text file can log in using the domain and name `INTERNAL\`<UserName> , where <*UserName*> is the registered user name.

The Internal user repository initially contains one predefined user named `Administrator` with the password `manage`. This user logs in using the domain and user name `INTERNAL\Administrator`. If your default authentication configuration is the Internal configuration, this user can log in using just the user name `Administrator`, without specifying the domain name explicitly.

You can perform administration tasks on the Internal authentication file, such as adding users, deleting users, and changing passwords.

CentraSite provides a command tool named `internaluserrepo` for this purpose.

You execute the command `internaluserrepo` in the command line tool `internaluserrepo.bat` (Windows) or `internaluserrepo.sh` (UNIX). The tool is located in the directory `<Software AG_directory>\common\bin\`. The internal authentication (.txt) file is located at `<Software AG_directory>\common\conf\users.txt` on all systems.

CAUTION:
As soon as possible after completing installation, you should change the password that is associated with the user `Administrator`.

The dialog for creating a configuration for Internal authentication prompts for the following values:

Parameter	Description
`Domain ID`	The domain ID is always `INTERNAL`. This cannot be changed.
`Expiration`	The number of seconds that the user is cached in the server after successful authentication. Changes made to the user, for example, deletion or password changes, do not take effect until this time has elapsed.

To modify the Internal authentication file

- Run the command `internaluserrepo`.

  The syntax is of the format:

  For more information about the usage of `internaluserrepo` command tool, see http://documentation.softwareag.com/webmethods/wmsuites/wmsuite9-7/Security_Infrastructure/9-7_Security_Infrastructure/using/Creating_Internal_User.htm.

Configuring LDAP Authentication Type

CentraSite supports various LDAP configurations and provides standard settings that allow you to set up your authentication quickly against these standard systems.

There are many questions that are involved when you configure against an LDAP system:

- What is the hierarchical node structure of the LDAP server?
- In which kinds of objects are the user and group definitions contained?
- Which node properties contain the user names or group IDs?
- What other property mappings are required?

In general, before you begin to specify the configuration, **Software AG** recommends you to study the LDAP structure and contents using an LDAP browser. There are various freeware tools that allow you to do this. Using the LDAP browser, you can bind to an LDAP server, then navigate through the hierarchy to see the structures that contains the users and groups. Also, you can open the nodes that contain the definitions of individual users or groups and view the properties that are stored for each user or group. An example of a node for a user `testuser01` might show the following properties:

Property name	Value
cn	testuser01
objectClass	OpenLDAPperson
Mail	JohnSmith@MyCompany.com
Phone	+1 234 555 678

The path to the node for this user might be `com/People/Location3/testuser01`, where `com` is the root node. The setup on this LDAP server might be that all users are stored under the `People` node (`com/People/…`) and all groups are stored under the `Groups` node (`com/Groups/…`). Since every CentraSite customer can define their LDAP user and group structures differently, the details of the LDAP configuration that you will perform in CentraSite vary accordingly, since you must map explicitly to the customer LDAP structures.

Technical Principal for LDAP

CentraSite can only find and authenticate a user name through the LDAP mechanism if either:

- the user name is located directly beneath the LDAP node that represents all users (specified through the User DN configuration value – for example, if user names are in the form `uid=Username,ou=people,dc=mydomain,dc=com` then the user name must be beneath the node `ou=people,dc=mydomain,dc=com`), or:
- the LDAP server allows anonymous bind.

The technical principal is a user name or user account that preferably should not belong to a real user, in other words, the technical principal is normally the ID of a fictitious user. It is intended for organizations that store their user entries in branched LDAP directory structures, for example `uid=Username,loc=Germany,ou=people,dc=mydomain,dc=com` but do not allow anonymous bind. The technical principal must be defined in LDAP as having (at least) read access to all users and groups that are to be used by CentraSite.

When CentraSite is configured to use this feature, *all* LDAP accesses take place using the technical principal. For example, if a user with user name `user1` and password `pwd1` wants to log in to CentraSite Control, LDAP is accessed using the technical principal and the record for the user `user1` is checked.

Configuring LDAP Authentication

Pre-requisites:

To add a new (LDAP) authentication configuration through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `set Authentication` for this purpose.

Beginning with version 9.10, the command tool has been enhanced to offer stricter validation. This validation behaviour allows CentraSite to simultaneously verify the newly added LDAP authentication configuration.

The `set Authentication` command opens an interactive dialog that prompts you to enter the basic details for LDAP authentication.

Note:
When executing the `set Authentication` command, CentraSite always resets some of your LDAP configuration properties to their default values. Such LDAP properties and their default values are summarized below:

- `alias` to `domain`

- applyDomain to true
- createGroupProperties totrue
- createGroups to true
- createUserProperties to true
- useaf to false
- usecaching to false

After executing the `set Authentication` command, any changes you make to these LDAP properties will be lost.

- Run the command `set Authentication`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Authentication -domain <DOMAIN>`

  Example (all in one line):

  ```
  C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Authentication -domain SAG
  ```

  The sample interactive dialog is as follows. During each step of the command, the server prompts you to enter the basic details for LDAP authentication.

```
==========================================================================
Step 1 - LDAP Server Configuration
--------------------------------------------------------------------------
Configuration Enabled (Y/N) [Y]:
URL of the LDAP server (ldap(s)://host:port): ldaps://ceres:10636
Connection Timeout (Milliseconds) [5000]:
Do you want to use the LDAP Technical user (Y/N) [N]: y
Principal (Technical User) DN: cn=techuser,ou=people,ou=gdm,o=sag
Password of Technical User:
Truststore Type (JKS, PKCS12): jks
URL of Truststore Location: file:/C:/TMP/trusted.ks
Truststore Password:
Keystore Type (JKS, PKCS12):
URL of Keystore Location:
Keystore Password:
--------------------------------------------------------------------------
Check 1 - Verifying LDAP Server Configuration. Please wait...
LDAP Server Configuration validated successfully.
Repeat configuration step, Continue, or End? (R/C/E) [C]:
==========================================================================
Step 2 - User Information Configuration
--------------------------------------------------------------------------
User Id. Attribute [cn]:
User Root DN (Location to be searched for users): ou=people,ou=gdm,o=sag
User Object Class [inetOrgPerson]:
User id. that is used to verify login: psinger
Password that is used to verify login:
--------------------------------------------------------------------------
Check 2 - Verifying User Configuration. Please wait...
User logged in successfully.
Search for user was successful.
User Configuration validated successfully.
Repeat configuration step, Continue, or End? (R/C/E) [C]:
==========================================================================
Step 3 - User Mapping Configuration
```

```
--------------------------------------------------------------------------------
emailAddresses:emailAddress:address       : mail
personName:firstName                      : givenName
personName:fullName                       : displayName
personName:lastName                       : sn
postalAddresses:postalAddress:postalCode  : postalCode
postalAddresses:postalAddress:streetNumber: postalAddress
telephoneNumbers:telephoneNumber:number   : telephoneNumber
Do you want to keep this mapping (Y/N) [Y]:
Search criteria to verify the search for users [cn=userid*]: cn=ino*
--------------------------------------------------------------------------------
Check 3 - Verifying User Mapping Configuration. Please wait...
The following attributes have been retrieved for user "psinger":
   displayName    : Peter Singer
   mail           : psinger@gdm.sag
   givenName      : Peter
   sn             : Singer
   telephoneNumber: +49 6151 92 0001
The following users match the search criteria "cn=ino*"
(only first ten are displayed):
   SAG\inosec1
   SAG\inosec10
   SAG\inosec2
   SAG\inosec3
   SAG\inosec4
   SAG\inosec5
   SAG\inosec6
   SAG\inosec7
   SAG\inosec8
   SAG\inosec9
   SAG\inotst
User Mapping Configuration validated successfully.
Repeat configuration step, Continue, or End? (R/C/E) [C]:
================================================================================
Step 4 - Group Information Configuration
--------------------------------------------------------------------------------
Group Id. Attribute [cn]:
Group Root DN (Location to be searched for groups): ou=groups,ou=gdm,o=sag
Group Object Class [group]: groupOfUniqueNames
Group id. that is used to verify settings: ManageAssets
--------------------------------------------------------------------------------
Check 4 - Verifying Group Configuration. Please wait...
Group Configuration validated successfully.
Repeat configuration step, Continue, or End? (R/C/E) [C]:
================================================================================
Step 5 - Group Mapping Configuration
--------------------------------------------------------------------------------
Please provide your LDAP attributes for groups
description: description
Search criteria to verify the search for groups [cn=groupid*]: cn=*
--------------------------------------------------------------------------------
Check 5 - Verifying Group Mapping Configuration. Please wait...
The following attributes have been retrieved for group "ManageAssets":
   description: manage assets
The following groups match the search criteria "cn=*"
(only first ten are displayed):
   SAG\Communiqu?
   SAG\FineGroup
   SAG\group1
   SAG\HighSearch
```

```
   SAG\inosecgroup
   SAG\invalidgroup
   SAG\ldadmingroup
   SAG\ldadmingroup1
   SAG\ldadmingroup2
   SAG\ldusergroup
   SAG\ManageAssets
Group Mapping Configuration validated successfully.
Repeat configuration step, Continue, or End? (R/C/E) [C]:
================================================================================
Step 6 - Group Resolution Configuration
--------------------------------------------------------------------------------
Membership Attribute is on Group Object (Y/N) [N]: y
Membership Attribute: uniqueMember
Recursive Depth for Group Search [0]: 1
--------------------------------------------------------------------------------
Check 6 - Group Resolution Configuration
User "psinger" belongs to the following groups:
   SAG\group1
   SAG\FineGroup
Group Resolution Configuration validated successfully.
Repeat configuration step, Continue, or End? (R/C/E) [C]:
================================================================================
Step 7 - Save Configuration
--------------------------------------------------------------------------------
Do you really want to save the configuration (Y/N): y
Configuration has been successfully saved.
Successfully executed the command : set Authentication
```

LDAP Authentication Configuration Parameters

The CentraSite Command `set Authentication` opens an interactive dialog that prompts you to enter the basic details for LDAP authentication.

The general values that you can specify for an LDAP authentication in the interactive dialog are described below.

Refer to the documentation of your LDAP system supplier for details.

LDAP Server Configuration

Prompt Text: Configuration Enabled

Description: Indicates if the LDAP server configuration is enabled or disabled.

Property: `enabled`

Value: The possible values are `true` and `false`.

Prompt Text: URL of the LDAP server

Description: The URL of the machine where the LDAP server is located. The expected format is:

- `ldap://<host>:<port>`
- `ldaps://<host>:<port>`

- To use an SSL connection for the LDAP server, you must specify the URL to start with `ldaps` and provide the truststore and keystore parameters.
- To use an IPv6 address (instead of the domain name), you must enclose the URL in square brackets.

Property: `url`

Value: `ldaps://<listening address>:10636`

Prompt Text: Connection Timeout

Description: The maximum time interval (in milliseconds) for an LDAP operation. The default value is 5000.

Property: `timeout`

Prompt Text: Principal (Technical User) DN

Description: The distinguished name (DN) of the technical user that connects to the LDAP server if an anonymous access to the LDAP server is not allowed.

For more background information on the technical user, see Technical Principal for LDAP.

Property: `prin`

Value: `cn=techuser,ou=people,dc=mydomain,dc=com`

Prompt Text: Password of Technical User

Description: The password for the technical user identified by the property `prin`.

This property is required only if the related property `noPrinIsAnonymous` is set to `false`. Otherwise, this property must not be specified.

Property: `@secure.cred`

Prompt Text: Truststore Type

Description: The type of truststore to use if an SSL connection is required.

Property: `truststoreType`

Value: The possible values are `JKS` and `PKCS12`.

Prompt Text: URL of Truststore Location

Description: The URL of the truststore containing the trusted root certificates.

Property: `truststoreUrl`

Value: `file:/C:/TMP/trusted.ks`

Prompt Text: Keystore Type

Description: The type of keystore to use if an SSL connection is required.

Property: `keystoreType`

Value: The possible values are `JKS` and `PKCS12`.

Prompt Text: URL of Keystore Location

Description: The URL of the keystore containing the private keys and SSL certificates.

Property: `keystoreUrl`

Value: `file:/C:/TMP/keystore.ks`

Prompt Text: Keystore Password

Description: The password of the keystore.

Property: `@secure.keystorePassword`

User Information Configuration

The user-specific settings that you can specify for an LDAP configuration are described below.

Prompt Text: User Id. Attribute

Description: Specifies the LDAP username attribute. This is the name of the property in the user node that is used to uniquely identify a user.

Property: `cn`

Value: `uidprop`

Prompt Text: User Root DN (Location to be searched for users)

Description: The location to search for users. The directory tree part of the distinguished name (standard LDAP terminology) of the entry. The method of specifying the path uses the standard LDAP path convention: first, a unique property of the DN node is specified, along with the property's value. Usually the property `ou` (organizational unit) is the property selected for this purpose. Then the next higher `dc` node (that is, a node with a `dc` property), then the next higher `dc` node and so on, until finally the root node.

Property: `userrootdn`

Value: `ou=people,dc=mydomain,dc=com`

Prompt Text: User Object Class

Description: Specifies that the identified object is a person and is used to categorize nodes as user nodes. The login module uses this parameter when searching for users.

Property: `inetOrgPerson`

Value: `personobjclass`

User Mapping Configuration

For background information on User Mapping Configuration, see "Mapping User and Group Fields" on page 73.

Group Information Configuration

The group-specific settings that you can specify for an LDAP configuration are described below.

Prompt Text: Group Id. Attribute

Description: Specifies the LDAP group attribute. This is the name of the property in the group node that is used to uniquely identify a group.

Property: `gidprop`

Value: `cn`

Prompt Text: Group Root DN (Location to be searched for groups)

Description: This is similar to the DN property for users, but identifies a DN node for groups rather than for users.

Property: `grouprootdn`

Value: `ou=groups,dc=mydomain,dc=com`

Prompt Text: Group Object Class

Description: Specifies that the identified object is a group and is used to categorize nodes as group nodes. The login module uses this parameter when searching for groups.

Property: `groupobjclass`

Value: `groupOfUniqueNames`

Group Mapping Configuration

For background information on Group Mapping Configuration, see "Mapping User and Group Fields" on page 73.

Group Resolution Configuration

Prompt Text: Membership Attribute is on Group Object

Description: Specifies whether the login module searches users in a group or groups in a user. The login module searches groups in a user.

Property: `memberinfoingroups`

Value: The possible values are:

- `Yes/true` - The login module searches users in a group.

- `No/false` - Default value.

Prompt Text: Membership Attribute

Description: The login module uses this parameter when performing member-search operations. The meaning of this parameter depends on the value of property `memberinfoingroups`.

Property: `memberinfoingroups`

Value: The possible values are:

- `true`: The property `mattr` points from a group to the users that are members of this group.
- `false`: The property `mattr` points from a user entry to the groups that the user is a member of.

Property: `mattr`

Value: `uniqueMember`

Prompt Text: Recursive Depth for Group Search

Description: Specifies the depth in the tree to which the nested groups should be searched.

Property: `recursiveSearchDepth`

Logging of Login Authentication Messages

If you have configured your authentication settings but still experience problems when trying to log in, you can use CentraSite's log files to analyze the problem. Some log file entries contain information about authentication problems in general, whereas other log file entries contain information about authentication problems related to individual CentraSite components.

You can activate the authentication logging by configuring the options in the CentraSite login context of `jaas.config` file. The options in the `jaas.config` file allow you to make the following changes:

- Switch authentication logging on or off for all CentraSite components.
- Specify the depth of logging required.

The CentraSite login context consists of one or more modules. Each individual module is defined by a specification. For example, you might specify a single login module `LDAPLoginModule` like the example shown below:

```
com.softwareag.security.sin.is.ldap.lm.LDAPLoginModule ...
```

You can specify arbitrary login modules. For example:

```
com.softwareag.security.sin.is.ldap.lm.LDAPLoginModule
com.softwareag.security.jaas.login.internal.InternalLoginModule
```

Options for activating the logging can be added to login modules: The available logging options are:

- **useLog.** Specify `true` to switch logging on, or `false` to switch logging off.

- **logLevel.** Specify the level of logging information required. Possible values are:
  - error - log only error messages
  - info - log error and information messages
  - debug - log all messages with additional debug information
- **logFile.** Specify the path and file name of the log file.

We recommend that you specify the logging options to the first occurrence of the above login modules.

- Open the file, jaas.config, in a rich text editor.

  You can find the file in the directory `<Software AG_directory>/profiles/CTP/configuration`.

  Example: To activate a SIN logging:

```
CentraSite {
  com.softwareag.security.sin.is.ldap.lm.LDAPLoginModule required
    useLog="true"
    logFile="/opt/softwareag/profiles/CTP/logs/sin-SAG-LDAP.log"
    logLevel="DEBUG"
    domain="SAG"
    alias="SAG"
    applyDomain="true"
    url="ldap://daeqarh01.eur.ad.sag:10389"
    prin="cn=LdapUser4CSAdmin,ou=people,ou=gdm,o=sag"
    cred="manage"
    usecaching="false"
    useaf="true"
    dnprefix="cn="
    dnsuffix=",ou=people,ou=gdm,o=sag"
    userrootdn="ou=people,ou=gdm,o=sag"
    uidprop="cn"
    personobjclass="inetOrgPerson"
    mattr="uniqueMember"
    memberinfoingroups="true"
    grouprootdn="ou=groups,ou=gdm,o=sag"
    gidprop="cn"
    groupobjclass="groupOfUniqueNames"
    creategroups="true"
    createGroupProperties="true"
    createUserProperties="true";
};
```

  This configuration creates a log file: `/opt/softwareag/profiles/CTP/logs/sin-SAG-LDAP.log`

  The log shows whether login attempts are successful or not, and indicates the user domain where CentraSite attempted to find the login user information, for example:

```
...Authenticator (<domain>, ...) was created successfully
...login of user <username> (domain: <domain>) was successful.
```

  If the authentication was not successful, a message such as the following is displayed:

```
Login of user <username> (host: <hostname>, port:<portnumber>) failed.
```

Transforming and Migrating Internal and LDAP Configuration Data

Pre-requisites:

To transform and migrate the Internal and LDAP configurations to the new JAAS configuration through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

When upgrading CentraSite 9.6 or earlier to version 10.0, you must transform and migrate the Internal and LDAP configurations from the old Registry or Repository to the new CentraSite JAAS configuration.

To resolve this problem, a mechanism is available to migrate the Internal and LDAP login configuration in CentraSite.

CentraSite provides a command tool named `generate JaasConfiguration` for this purpose.

The tool generates the InternalLoginModule and LDAPLoginModule entries that correspond to the old Internal and LDAP configurations and saves the entries in the jaas.config file in the folder `Software AG_directory/profiles/CTP/configuration`. For each configured LDAP domain, the script creates user and group files that map internal (CentraSite) properties to external (LDAP) properties and saves the files in the `Software AG_directory/profiles/CTP/configuration/com.softwareag.platform.config.propsloader` directory.

The sample Jaas configuration file is as follows.

```
CentraSite {
    com.softwareag.security.jaas.login.internal.InternalLoginModule sufficient
        domain="INTERNAL"
        alias="INTERNAL"
        applyDomain="true"
        create_group_principal="false"
        internalRepository="C:/SoftwareAG/common/conf/users.txt";

    com.softwareag.security.sin.is.ldap.lm.LDAPLoginModule required
        domain="EUR"
        url="ldap://ldap-server:389"
        createGroupProperties="true"
        creategroups="true"
        dnprefix="cn="
        noPrinIsAnonymous="false"
        usecaching="false"
        alias="EUR"
        personobjclass="inetOrgPerson"
        useaf="true"
        grouprootdn="DC=EUR,DC=example,DC=com"
        userrootdn="DC=EUR,DC=example,DC=com"
        memberinfoingroups="false"
        dnsuffix=",ou=user,OU=Germany,DC=EUR,DC=example,DC=com"
        applyDomain="true"
        createUserProperties="true"
        groupobjclass="group"
        uidprop="sAMAccountName";
};
```

Note:
The `generate JaasConfiguration` command transforms only domains of type Internal and LDAP. If you have advanced JAAS configurations such as single-sign-on configurations, you must migrate them manually.

If you start the command line tool with no parameters, you will receive a help text summarizing the required input parameters.

If you omit the passwords from the command, you are prompted to provide them.

To transform and migrate Internal and LDAP configurations to JAAS configuration

- Run the command `generate JaasConfiguration`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd generate JaasConfiguration [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD>`

  The input parameters are:

Parameter	Description
`CENTRASITE-URL`	(Optional). The URL of the CentraSite registry. Default value is `http://localhost:53307`.
`USER-ID`	The user ID of a registered CentraSite user. For example, a user who has the CentraSite Administrator role.
`PASSWORD`	The password for the registered CentraSite user identified by the parameter `USER-ID`.

  Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd generate JaasConfiguration
-url http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
```

  The response to this command could be:

```
Executing the command : generate JaasConfiguration
Successfully executed the command : generate JaasConfiguration
```

Creating Technical User for Reconfiguring Migrated Configuration

Upgraded configurations might suffer from the limitation that the new LDAPLoginModule requires a technical user for dealing with incomplete user DNs. A missing technical user is indicated by the following error messages in the SIN.log:

Sample A

```
[LDAP: error code 49 - 80090308: LdapErr: DSID-0C0903A9, comment:
AcceptSecurityContext error, data 52e, v1db1]
```

Sample B

```
[LDAP: error code 49 - 80090308: LdapErr: DSID-0C0903AA, comment:
AcceptSecurityContext error, data 525, v1772]
```

The above error messages indicate that there was an authentication failure while attempting to login the user.

Important:
To create a SIN log, the following property lines need only be applied to the first occurring login module in the CentraSite login context of jaas.config file. You can find the file in the directory `<Software AG_directory>/profiles/CTP/configuration`.

```
useLog="true"
logFile="path-to-log-folder/SIN.log"
logLevel="DEBUG"
```

To configure a technical user, you must manually update the jaas.config file in the following way:

- Specify the full User DN value of the technical user in the `prin` property.

  ```
  prin="CN=tech-user,OU=Generic,OU=Germany,DC=eur,DC=ad,DC=sag"
  ```

- Specify the password of the technical user in the `cred` property.

  ```
  cred="password"
  ```

After making the above changes, the Jaas configuration would look like the following:

```
CentraSite {
    com.softwareag.security.jaas.login.internal.InternalLoginModule sufficient
        domain="INTERNAL"
        alias="INTERNAL"
        applyDomain="true"
        create_group_principal="false"
        internalRepository="C:/SoftwareAG/common/conf/users.txt";

    com.softwareag.security.sin.is.ldap.lm.LDAPLoginModule required
        domain="EUR"
        url="ldap://ldap-server:389"
        createGroupProperties="true"
        creategroups="true"
        noPrinIsAnonymous="false"
        prin="CN=tech-user,ou=user,OU=Germany,DC=EUR,DC=example,DC=com"
        cred="password"
        usecaching="false"
        alias="EUR"
        personobjclass="inetOrgPerson"
        useaf="false"
        grouprootdn="DC=EUR,DC=example,DC=com"
        userrootdn="DC=EUR,DC=example,DC=com"
        memberinfoingroups="false"
        applyDomain="true"
        createUserProperties="true"
        groupobjclass="group"
        uidprop="sAMAccountName";
};
```

Securing Login Information of Technical User

In the Jaas configuration file, the technical user password is stored in clear text.

You can find the jaas.config file in the directory `<Software AG_directory>/profiles/CTP/configuration`.

To secure the password, you must reconfigure the Jass configuration to reference the necessary LDAP configuration properties file, for example, in the following way:

```
CentraSite {
    com.softwareag.security.jaas.login.internal.InternalLoginModule sufficient
        domain="INTERNAL"
        alias="INTERNAL"
        applyDomain="true"
        create_group_principal="false"
        internalRepository="C:/SoftwareAG/common/conf/users.txt";

    com.softwareag.security.sin.is.ldap.lm.LDAPLoginModule required
        alias="EUR";
};
```

The LDPAPLoginModule entry in the Jaas configuration file no longer contains the LDAP configuration properties. Instead, the entry contains an alias parameter used to point to an EUR configuration. For defining the EUR configuration, the file `com.softwareag.security.ldap.server.pid-EUR.properties` must be stored in the folder `com.softwareag.platform.config.propsloader` under *<RuntimeDir>*/configuration.

The LDAP configuration properties file looks as follows:

```
alias=EUR
url=ldap\://ldap-server\:389
prin=CN\=tech-user,ou\=user,OU\=Germany,DC\=EUR,DC\=example,DC\=com
@secure.cred=password
useaf=false
userrootdn=DC\=EUR,DC\=example,DC\=com
personobjclass=inetOrgPerson
uidprop=samAccountName
noPrinIsAnonymous=false
groupobjclass=group
usecaching=false
applyDomain=true
createGroupProperties=true
memberinfoingroups=false
creategroups=true
createUserProperties=true
grouprootdn=DC\=EUR,DC\=example,DC\=com
domain=EUR
```

The property `@secure.cred` contains the `password` in clear text.

Software AG Runtime scans for new files and replaces the clear text password with a handle after securely storing the password in another location.

The dynamically loaded LDAP configuration properties file looks as follows:

```
###Fri Dec 19 14:02:42 CET 2014
useaf=false
userrootdn=DC\=EUR,DC\=example,DC\=com
personobjclass=inetOrgPerson
```

```
prin=CN\=tech-user,ou\=user,OU\=Germany,DC\=EUR,DC\=example,DC\=com
uidprop=samAccountName
url=ldap\://ldap-server\:389
noPrinIsAnonymous=false
groupobjclass=group
usecaching=false
applyDomain=true
@secure.cred=@secure\:com.softwareag.security.ldap.server.pid-
EUR.properties-cred
createGroupProperties=true
alias=EUR
memberinfoingroups=false
creategroups=true
createUserProperties=true
grouprootdn=DC\=EUR,DC\=example,DC\=com
domain=EUR
```

Here, the property `@secure.cred` does not contain the `password` in clear text. Instead, it contains a pointer to the internal secure password store. Thus the clear text password is replaced with the dynamic LDAP configuration as long as the Software AG Runtime is running.

Configuring Email Server

Certain facilities within CentraSite communicate information to users using email (the Send Email Notification policy action, for example). These facilities do not function properly until you configure CentraSite's email settings. These settings specify the Simple Mail Transport Protocol (SMTP) server that CentraSite is to use for sending outgoing email messages.

Configuring Email Server Settings

Pre-requisites:

To configure the email server settings through the Command Line Interface, you must have the CentraSite Administrator role.

In addition, you must know the name (or IP address) of the email server that CentraSite is to use and the port number on which that server listens for SMTP requests. If the email server is configured to authenticate users, you must additionally provide the user ID and password that CentraSite is to use to log on to the server.

CentraSite provides a command tool named `set Email` for this purpose.

To configure the email server settings

1. Create an XML configuration file (in Java XML properties format) with a set of predefined properties.

   Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
 <properties version="1.0">
 <entry key="com.centrasite.config.email.SMTPHost">localhost
```

```
 </entry>
 <entry key="com.centrasite.config.email.SMTPPort">25</entry>
 <entry key="com.centrasite.config.email.ReplyTo">
            noreply@editthisdomain.com
 </entry>
 <entry key="com.centrasite.config.email.ConnectionTimeout">20
 </entry>
 <entry key="com.centrasite.config.email.Authentication">false
 </entry>
 <entry key="com.centrasite.config.email.User">xyz
 </entry>
 <entry key="com.centrasite.config.email.Password">xyz
 </entry>
 <entry key="com.centrasite.config.email.TransportLayerSecurity">
            false
 </entry>
</properties>
```

The predefined properties are as follows:

In this property...	Specify...
SMTPHost	The name or IP address of the machine on which the SMTP server is running.
SMTPPort	The port on which the machine specified in **SMTP Host** listens for SMTP requests.
ReplyTo	The email address to which responses to the emails sent by CentraSite are to be directed. CentraSite uses this address to populate the From email header in the emails that it sends.
ConnectionTimeout	The number of seconds that CentraSite waits for the email server to accept a connection request. If the email server does not respond within the specified time period, CentraSite writes an error message to the console and discards the email.
Authentication	Whether the SMTP server authenticates users. Set **Authentication** to `yes` if authentication is enabled on the SMTP server. If you set this field to `yes`, you must specify the appropriate log-on credentials in the **User** and **Password** fields below.
User	The user ID that CentraSite is to use to log on to the SMTP server. This value is required only when **Authentication** is set to `yes`.
Password	The password that CentraSite is to use when it logs on to the SMTP server. This value is required only when **Authentication** is set to `yes`.

In this property...	Specify...
TransportLayerSecurity	If true, enables the use of the `STARTTLS` command (if supported by the server) to switch the connection to a TLS-protected connection before issuing any login commands. Note that an appropriate trust store must be configured so that the client trusts the server's certificate. Defaults to false.

2. Run the command `set Email`.

   The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Email [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -file <CONFIG-FILE>`

   The input parameters are:

Parameter	Description
`CENTRASITE-URL`	(Optional). The URL of the CentraSite registry. Default value is `http://localhost:53307`.
`USER-ID`	The user ID of a registered CentraSite user. For example, a user who has the CentraSite Administrator role.
`PASSWORD`	The password for the registered CentraSite user identified by the parameter `USER-ID`.
`CONFIG-FILE`	The absolute or relative path to the XML configuration file. If relative, the path should be relative to the location from where the command is executed.

   Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Email -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-file config.xml
```

   The response to this command could be:

```
Executing the command : set Email

Successfully executed the command : set Email
```

Fetching Email Server Settings

Pre-requisites:

To fetch the details of the email server settings through the Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `get Email` for this purpose.

To fetch the email server settings

1. Run the command `get Email`.

   The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get Email [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -file <CONFIG-FILE>`

   The input parameters are:

Parameter	Description
`CENTRASITE-URL`	(Optional). The URL of the CentraSite registry. Default value is `http://localhost:53307`.
`USER-ID`	The user ID of a registered CentraSite user. For example, a user who has the CentraSite Administrator's role.
`PASSWORD`	The password for the registered CentraSite user identified by the parameter `USER-ID`.
`PASSWORD`	The absolute or relative path to the XML configuration file. If relative, the path should be relative to the location from where the command is executed.

   Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get Email -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-file config.xml
```

   The response to this command could be:

```
Executing the command : get Email
Successfully executed the command : get Email
```

Implementing CentraSite

Before implementing CentraSite, define the goals and principles that you want to achieve by establishing SOA governance. Some of the questions to be considered before implementation are:

- Who are the stakeholders and what organizations do you want to reach with your SOA governance initiative?
- What are the stakeholders' role in the development of SOA artifacts?
- What are the governance rules and processes that you want to establish?
- What kind of artifacts do you want to manage utilizing CentraSite?
- How can you support better reuse by introducing company-wide classification of schemes for assets?

- How will you measure the success of your SOA environment and your SOA governance initiatives?
- What is the development life cycle of your artifacts today and what should it be? What type of stakeholder interactions occur during the life cycle transitions?

Configuration Checklist

The following list identifies the key configuration decisions and tasks that you need to do to use your CentraSite implementation:

Task	See...
Determine whether you may implement a single registry or multiple registries.	"Choosing a Deployment Strategy" on page 95
Install each instance of CentraSite and configure its connection to the external authentication system.	"Configuring User Authentication and Repositories" on page 65
Identify the organization structure you will use for each registry, identify organization administrators, and create the organizations.	For details on organization management, see *CentraSite User's Guide*.
Determine whether the default role assignments given to the Users group are appropriate for each organization and modify these assignments if necessary.	For details on user and role management, see *CentraSite User's Guide*.
Identify the types of assets that you want to catalog in CentraSite. Define new types of assets and customize the predefined types. Create the required taxonomies and association types to support your customizations.	For details on asset management, see *CentraSite User's Guide*.
Determine which asset types you want to place under lifecycle management and create the required lifecycle models.	For details on life cycle management, see *CentraSite User's Guide*.
Determine which polices you want to place on the objects in your registry and create the required design/change-time policies to enforce these policies.	For details on design/change-time policies, see *CentraSite User's Guide*.
Create the Consumer Registration policy on each instance of CentraSite.	For details on policy and consumer management, see *CentraSite User's Guide*.

If you are using CentraSite with webMethods Mediator for run-time mediation, you must also perform the following tasks to prepare CentraSite for use:

Task	See...
Define a gateway of type `Mediator` and a user account for each webMethods Mediator that is attached to an instance of CentraSite.	For details on gateway management, see *CentraSite User's Guide*.
Create the lifecycle model and associated policies required to enable deployment of virtual services.	For details on life cycle management, see *CentraSite User's Guide*.
Define the process your site uses for creating, deploying, and promoting virtual services and ensure that the participants in this process have the necessary permissions to perform their assigned tasks.	For details on policy management, see *CentraSite User's Guide*.

Choosing a Deployment Strategy

When planning your CentraSite implementation, you have to select a deployment strategy that supports your organization's SDLC and its implementation requirements. The strategy you select determines the number of stages (instances of a CentraSite registry) that your organization maintains and how it maps the phases of its SDLC to these stages.

Deploying to Support your SDLC

When developing Web services and other assets, most IT organizations follow a Systems Development Life Cycle (SDLC) that includes the following basic phases:

- *Development phase* - Web services and other assets are requested and developed, individual contributions are integrated and development tests are conducted.
- *Test phase* - Services and other assets are tested in a controlled environment that simulates production scenarios.
- *Production phase* - Services and other assets are made operational. When an asset reaches this stage, the environment in which it resides is tightly controlled and access to the asset is restricted.

In CentraSite, the steps of a SDLC are represented by a *lifecycle model*. The lifecycle model is customized for your environment and enables you to establish governance controls over all phases of the SDLC for different types of assets.

Deployment Options

There are three basic deployment options. Review the following options and select the strategy that best supports your organization's SDLC and its governance objectives:

- **Single-Stage Deployment** - The entire SDLC is represented within one instance of CentraSite. You deploy and maintain a single registry. The assets you place in the registry remain in the registry over their entire lifecycle.

When you use single-stage strategy, you map all three basic phases of the SDLC to a single lifecycle model in CentraSite. To promote an asset through the SDLC, you switch the asset's lifecycle state in the registry. For example, moving an asset from the development phase to the test phase.

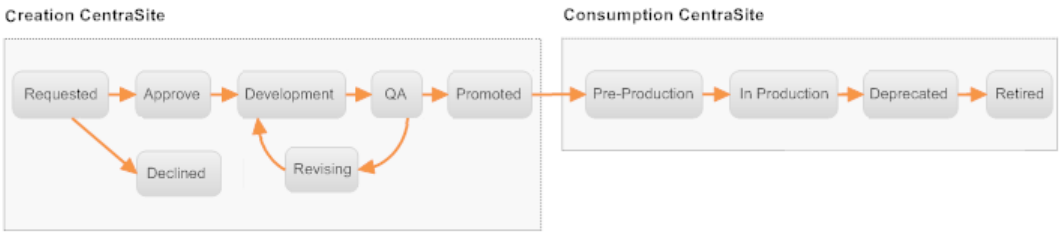
The following figure shows single-stage deployment:

- **Two-Stage Deployment** - In a two-stage deployment, the SDLC is split between two instances of CentraSite. One instance, called the creation CentraSite, is used to manage assets during the development and test phases of the SDLC. The other instance, called the consumption CentraSite, manages assets that are in the production phase of SDLC.

  This strategy enables your organization to completely separate assets in the pre-production phases from assets that are actually operational. In some organizations, the physical separation of development and production systems is necessary to satisfy legal regulations.

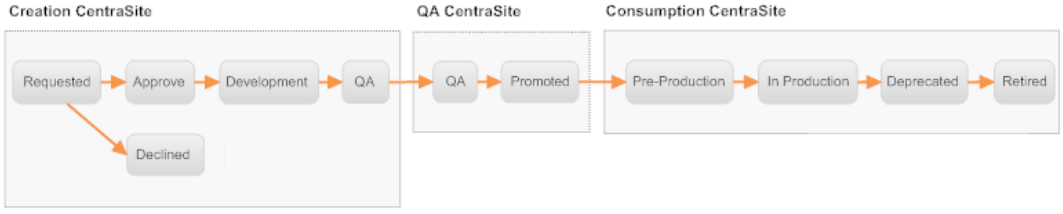
  When you use the two-stage approach, SDLC is represented by two lifecycle models in CentraSite. One lifecycle model exists on the creation CentraSite. This model represents the states that constitute the development and test phases of the SDLC. The other model exists on the consumption CentraSite and represents the states that constitute the production phase of the SDLC.

  To promote an asset to a phase of its lifecycle that resides on another stage, you export the asset from its current registry and import it into the registry that hosts the next phase of its lifecycle.



- **Three-Stage Deployment** - In a three-stage deployment, you deploy a separate registry for each major phase of the SDLC: Development, Test, and Production.

  Each registry has a lifecycle model that represents the states that constitutes its phase of the SDLC. You can promote an asset from one phase to the next by exporting the asset from its current registry and importing it into the registry that hosts the next phase of the SDLC.



Deployment Considerations

The deployment strategy you select depends on factors such as your organization's policy requirements, standard processes, and governance objectives. Consider the following when selecting a strategy for your organization:

- Any deployment that involves multiple stages requires additional effort to configure and administer. The promotion process for a multi-stage environment is more complex and time-consuming as it involves physically exporting objects from one registry and importing them into another. You should not deploy a multi-stage configuration unless there is a compelling reason to do so. Aim for a deployment strategy that aligns well with your organization's SDLC process, satisfies your organization's governance objectives, and uses the fewest number of stages.
- If you intend to use CentraSite for both design-time governance and run-time governance, consider using the two-stage deployment. This configuration enables you to maintain one registry for managing Web services (and associated assets) while they are in the development and testing phases of their lifecycle and another registry for configuring, deploying, and monitoring Web services that are in the production phase of their lifecycle.

  Although it is possible to use a single-stage deployment for both design-time and run-time governance, such a configuration is suitable only for small or mid-size environments. Do not use a single-stage deployment if you intend to use CentraSite to manage both the design-time and run-time aspects of a large number of assets.

  Note:
  If you expect your organization's registry to begin small and grow over time, start with a two-stage deployment rather than attempting to switch when you outgrow the single-stage configuration.
- If you intend to use CentraSite only for design-time governance, consider using a single-stage deployment. Deploying a multi-stage configuration for design-time governance does not offer any benefits. You should only consider a multi-stage deployment in a design-time implementation, if your organization has a specific need to physically separate the registry of assets in the pre-production phases from the registry of assets in the production phase.

Implementing the Mediation Environment

To use an instance of CentraSite with webMethods Mediator, you must define a gateway that identifies the specific Mediator that you want to use. A gateway is a registry object that represents a particular instance of a policy enforcement point (in this case, an instance of webMethods Mediator). The target object specifies the address of the Mediator's deployment endpoint, which is the endpoint that CentraSite uses to interact with Mediator to deploy virtual service.

If you use multiple Mediators with an instance of CentraSite, you must create a gateway for each Mediator. To easily distinguish Mediators when they are viewed in CentraSite, consider adopting a naming convention for gateways that clearly identifies to which environment the target belongs (for example, development, test, production).

To communicate with CentraSite, Mediator must have a user account on CentraSite. You have to establish this user account before you can configure Mediator's connection to CentraSite. Because Mediator reads and writes to certain objects in the registry (such as virtual services), its user account requires a specific set of permissions. These permissions are explained in detail in the Mediator documentation.

Note:
If you expect a high volume of traffic through a particular Mediator, consider clustering that Mediator. When you cluster Mediators, the cluster is represented by a single gateway within CentraSite.

Managing the Collection of Metrics

webMethods Mediator collects performance data (for example, average response time, total request count, fault count) for the virtual services that it hosts. It publishes this data to CentraSite at regular intervals. When you install and configure Mediator, you must specify whether you want it to collect performance data and, if so, how often you want it to publish the data to CentraSite.

Software AG recommends that you always enable the collection of performance data on your Mediator. A publication interval of 15 minutes is appropriate for most environments. However, if Mediator handles a very high volume of traffic, consider increasing this interval to 30 or 60 minutes. CentraSite stores the performance data that it receives from the Mediator in the performance log. You can look at the performance information for a particular virtual service by viewing the virtual service's **Performance** profile.

The performance data that CentraSite collects from Mediator can cause the log to grow quite rapidly. When the log grows very large, queries to the log can significantly affect CentraSite's performance. To prevent this from happening, you have to routinely purge old entries from the log.

CentraSite provides a log-purging utility that you can use to automatically purge the log on a scheduled basis. You can use this utility to select a time interval for which you may want to retain the data. When you configure the log-purging utility, you can specify whether you want to delete the purged log entries or export them to an archive file for future reference.

Note:
The performance metrics that Mediator collects enable service consumers (and potential service consumers) to determine whether a virtual service is performing at a required level. However, Mediator does not collect data at the granularity that a network administrator would need in order to analyze performance problems (for example, to determine why the response time for a particular virtual service drops at a particular time of day).

Managing the Collection of Events

In addition to performance metrics, webMethods Mediator can also log *event data*. Event data supplies information about activities or conditions that occur on Mediator.

Mediator logs two basic kinds of events:

- Data related to the operation of Mediator

- Data related to the execution of virtual services

The event data that Mediator collects about itself are referred to as *lifecycle events*. These events represent activities or conditions that occur during the general operation of Mediator. Lifecycle events are reported for the completion of significant processes (for example, Mediator start-up) and the detection of operational exceptions and policy violations. Mediator logs lifecycle events if you configure it to do so.

Mediator collects the following types of event data relating to the execution of virtual services. Mediator does not collect this type of information automatically. If you want to capture these types of events, you must deploy run-time policies to do so:

- *Transaction Events* report information about the requests that the Mediator processes. This type of event is produced by the execution of a logging action in a run-time policy. For example, you might configure a run-time policy to log all the request and response messages submitted to a particular virtual service.
- *Monitoring Events* report transgressions relating to performance metrics. This type of event is produced by the execution of a monitoring action in a run-time policy. For example, you might configure a run-time policy to report occasions when the response time for a virtual service exceeds a specified threshold.

Mediator publishes event data as Simple Network Management Protocol (SNMP) traps. When you install Mediator, you can configure it to publish this data to CentraSite, to another third-party SNMP server or to both. If you choose to log event data to CentraSite, you can view the events using the CentraSite Control user interface.

Note:
If you are logging event data to CentraSite, you must first configure CentraSite's event receiver as described in CentraSite User's Guide.

Like the performance log, the event log grows larger over time. If it becomes very large, queries to the log cause performance issues with CentraSite. To manage the size of the event log, you have to occasionally purge old entries from it. As a general guideline, consider maintaining only three months of event data in the log or as required. For example, if you routinely log request and response messages you might need to purge more often.

You can configure CentraSite's log-purging utility to purge the event log on a scheduled basis. When you configure the purging facility, you can specify whether you want to delete the purged entries or export them to an archive file for future reference.

Using CentraSite with Other Policy Enforcement Points

Instead of (or in addition to) using webMethods Mediator for mediation and policy enforcement, you can use other third-party products with CentraSite. Support for third-party policy-enforcement and run-time governance tools is available through integrations that are provided by members of the Software AG Community. These tools are made available through the Software AG Community Web site at Software AG TECHcommunity Website .

2 Managing Logs

- Overview 102
- Configuring Logs 103
- Obtaining Log Configuration Settings 107
- Monitoring Logs 108
- Audit Logs for User Information 108
- Purging Logs 115
- Exporting the Purged Log Records 119
- Configuring Purger Properties for High Volume Data Handling 119
- Removing Leftover Auditable Events 121

Overview

To effectively manage the CentraSite registry or repository, it is necessary to get feedback about the activity and performance of CentraSite. CentraSite provides comprehensive and flexible logging functionality for tracking design/change-time and run-time events. CentraSite stores the log and monitoring data of all registry objects (for example, policies, assets) as log records in the CentraSite Registry Repository (CRR).

As your log records increases, you may want to purge log records to avoid performance degradation. CentraSite provides the ability to purge log records and back them up to another location. You can purge log records manually (on demand) or automatically on a scheduled basis.

Logging Data

CentraSite's Logging functionality provides the ability to gather and analyze the following information:

- Run-time performance data posted by a gateway (that is, a policy enforcement point (PEP) or a run-time monitoring component such as Insight).
- Run-time event data (transaction events, policy violation events, and so on) posted by a gateway.
- Approval history data.
- Design/change-time policy data.
- Audit data.
- Consumer Registration data.

CentraSite includes the following system logs:

- **Policy Log:** The Policy Log contains information about the design/change-time policies that CentraSite has executed. By default, CentraSite only logs information about policies that fail. However, you can optionally configure CentraSite to log information about policies that resulted in success, informational, warning, and failure alerts.

  In addition, if you purge the policy log, CentraSite makes an entry in the policy log to indicate that entries have been purged.
- **Approval History Log:** The Approval History Log contains a record of all approval requests that have been triggered by a policy with an approval workflow action. This log shows the status of each approval request that has been submitted to CentraSite.
- **Audit Log:** An Audit Log reports on the creation and update activities of a particular asset (including changes in an asset's lifecycle state).
- **Run-Time Event Log:** The Run-Time Event Log contains information about run-time events that have occurred in a gateway (that is, a policy-enforcement point (PEP) or a run-time monitoring component).
- **Run-Time Performance Log:** The Run-Time Performance Log captures run-time metrics for all assets and publishes them to CentraSite at regular intervals.

Purging Data

The term obsolete refers to any data not needed by the database but still occupying space on it. This unwanted or obsolete data can eventually fill up the disk and decrease the database performance, causing time-consuming database lookups, contention issues and so on. The process of systematically removing this unwanted data from the database is called *purging*. This process basically involves running the Purger scripts. They are available for all log units stored in the CentraSite Registry Repository.

Typically, you need the CentraSite Registry Repository to include only the recent log records. For instance, each CentraSite object has a number of actions performed on it, and for each action a separate audit log is generated and stored in the CentraSite Registry Repository. This increases the size of the CentraSite Registry Repository dramatically. Failure to purge the excess log records in the database could cause problems. Therefore, the purge interval must be configured as required.

Each time you purge a log unit, the corresponding purging log entry is created and can be viewed in CentraSite Control. For example, when you purge a certain set of policy logs, the corresponding purging log entry is automatically created and is visible in the Policy Log page.

Configuring Logs

Pre-requisites:

To configure the log settings, you must have the CentraSite Registry Repository up and running.

Configuring the lot settings is a two-step process:

1. Create a log configuration (`config.xml`) file.
2. Execute the script file from a command line with appropriate input parameters.

Configuring Log Settings

Pre-requisites:

To configure the log settings for specified log units through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `set Log` for this purpose.

To configure the log settings

- Run the command `set Log`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Log [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -file <CONFIG-FILE>`

  The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, http://localhost:53307/CentraSite/CentraSite.
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, Administrator.
PASSWORD	The password for the registered CentraSite user identified by the parameter USER-ID.
CONFIG-FILE	The absolute or relative path to the XML configuration file. If relative, the path must be relative to the location from where the command is executed.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Log -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-file config.xml
```

The response to this command could be:

```
Executing the command : set Log
Successfully executed the command : set Log
```

Creating Log Setting Configuration File

You create a log configuration file, config.xml, to define the log settings.

Ensure that the log configuration file contains the XML namespaces for providing uniquely named elements and attributes.

Defining a Log Unit

The unique element LogUnit represents the com.centrasite.config.log.unit property in config.xml. The LogUnit element specifies the log record to be stored in the CentraSite Registry Repository.

In the LogUnit element, add an attribute called name and assign a value to it which is equal to either of the following:

Log Unit	Description
Policy_Log	Contains information about the design/change-time policies that CentraSite has executed.
Approval_Log	Contains information about all approval requests that has been triggered by a policy with an approval workflow action.

Log Unit	Description
`Audit_Log`	Contains information on the creation or update activities of a particular asset (including changes in an asset's lifecycle state).
`Runtime_Event_Log`	Contains information about run-time events that have occurred in a gateway (that is, a policy-enforcement point (PEP) or a run-time monitoring component).
`Runtime_Performance_Log`	Contains information about the KPI metrics of all rogue assets.

This `LogUnit` element must look similar to this:

```
<LogUnit name=Policy_Log></LogUnit>
```

Defining the Log Settings

The `com.centrasite.config.log.setting` property represents the `LogSetting` attribute. This attribute defines specific information about the performance of the registry object.

In this section, you create log settings that is used within the log unit definition.

Within the `LogUnit` element, create `LogSetting` attributes say, `Success` and, `Failure`.

You can configure the `LogUnit` element with one of the appropriate `LogSetting` attributes.

Log Unit	Description	
Policy_Log	**Log Setting**	**Logs**
	Success	Policies that have resulted in success alert.
	Info	Policies that have resulted in informational alert.
	Warning	Policies that have resulted in warning alert
	Failure	Policies that have resulted in failure alert.
Approval_Log	**Log Setting**	**Logs**
	Log_Approval	Approval policies that have resulted in Approved state.
Audit_Log	**Log Setting**	**Logs**

Log Unit	Description	
	Enable_Audit	All activities (such as creation, modified, and so on) performed on a particular asset.
Runtime_Event_Log	**Log Setting**	**Logs**
	Policy_Violation_Events	Monitors and tracks all the policy violation events.
	Transaction_Events	Monitors and tracks all the transaction events.
	Monitoring_Events	Monitors and tracks all the monitoring events.
	Lifecycle_Events	Monitors and tracks all the lifecycle events.
Runtime_Performance_Log	**Log Setting**	**Logs**
	Log_Performance_Events	Monitors and tracks all the performance events

This `LogSetting` attribute must look similar to this:

```
<LogUnit name=Policy_Log>
    <LogSetting>Success</LogSetting>
    <LogSetting>Failure</LogSetting>
</LogUnit>
```

The final `config.xml` must look similar to this:

```
<LogSettings xmlns="http://namespaces.centrasite.com/configurations/logs">
    <LogUnit name=Policy_Log>
        <LogSetting>Success</LogSetting>
    </LogUnit>
    <LogUnit name=Approval_Log>
        <LogSetting>Log_Approval</LogSetting>
    </LogUnit>
    <LogUnit name=Audit_Log>
        <LogSetting>Enable_Audit</LogSetting>
    </LogUnit>
    <LogUnit name=Runtime_Event_Log>
        <LogSetting>Transaction_Events</LogSetting>
        <LogSetting>Policy_Violation_Events</LogSetting>
        <LogSetting>Monitoring_Events</LogSetting>
        <LogSetting>Lifecycle_Events</LogSetting>
    </LogUnit>
    <LogUnit name=Runtime_Performance_Log>
        <LogSetting>Log_Performance_Events</LogSetting>
    </LogUnit>
</LogSettings>
```

If you do not specify a log setting value or if you specify an incorrect log unit or log setting value (for example, `Log Approval`), the command execution fails and a warning message is issued.

```
(this is correct)
<LogUnit name=Policy_Log>
  <LogSetting>Success</LogSetting>
</LogUnit>
```

```
(this is incorrect due to missing LogSetting element)
<LogUnit name=Policy_Log>
</LogUnit>
```

```
(this is incorrect due to invalid LogSetting value)
<LogUnit name=Policy_Log>
  <LogSetting>Success_and_Failure</LogSetting>
</LogUnit>
```

Note:
Make sure you copy this file somewhere within the file system of the machine where CentraSite is installed.

Obtaining Log Configuration Settings

Pre-requisites:

To obtain the details of the log configuration settings through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `get Log` for this purpose.

To obtain log configuration settings

- Run the command `get Log`.

  The syntax is of the format:`C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get Log [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -file <CONFIG-FILE>`

  The input parameters are:

Parameter	Description
`CENTRASITE-URL`	(Optional). The URL of the CentraSite registry. For example, `http://localhost:53307/CentraSite/CentraSite`.
`USER-ID`	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, `Administrator`.
`PASSWORD`	The password for the registered CentraSite user identified by the parameter `USER-ID`.
`CONFIG-FILE`	The absolute or relative path to the XML configuration file. If relative, the path must be relative to the location from where the command is executed.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get Log -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-file config.xml
```

The response to this command could be:

```
Executing the command : get Log
Successfully executed the command : get Log
```

Monitoring Logs

After you have configured CentraSite to log event messages, you can monitor the events by viewing these log messages. By examining the log files you can monitor various aspects of the CentraSite's design/change-time and run-time events.

For more information about monitoring logs across different services, see the *CentraSite User's Guide*.

Audit Logs for User Information

CentraSite uses the system-defined log settings to store user information. The following user activities are covered:

- Start of user session
- End of user session
- Database updates

Starting the Audit Logs

To start the logging of CentraSite user information through the Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `start auditLog` for this purpose.

To start the logging of CentraSite user information

- Run the command `start auditLog`.

  The syntax is of the format `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd start auditLog [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD>`

  The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, http://localhost:53307/CentraSite/CentraSite.
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, Administrator.
PASSWORD	The password for the registered CentraSite user.

Example:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd start auditLog -user
Administrator -password manage -url http://localhost:53307/CentraSite/CentraSite
```

The response to this command could be:

```
Executing the command : start auditLog

Successfully executed the command : start auditLog
```

Storing the Audit Log Data

To store the audit log of user information through the Command Line Interface, you must have the CentraSite Administrator role.

When you use the `start auditLog` command, CentraSite logs the information in the current audit log.

In certain circumstances, you may want to read the audit log and display the information as a standard output or write the information to a file.

CentraSite provides a command tool named `unload auditLog` for this purpose.

To start the logging of CentraSite user information

- Run the command `unload auditLog`.

  The syntax is of the format `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd unload auditLog [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> [-outputFile <LOGDATA-FILE>] [-logNumber <LOG-NUMBER>]`

  The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, http://localhost:53307/CentraSite/CentraSite.
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, Administrator.

Parameter	Description
PASSWORD	The password for the registered CentraSite user.
LOGDATA-FILE	(Optional). The name of the file to which the data is to be written.
LOG-NUMBER	(Optional). The log file number to be processed. **Note:** If you are starting the audit log for the first time, the log entries are written to audit log 1. You can select to write the log entries to the next audit log file using the `switch auditLog` command.

Example (UNIX):

```
odsinst@daeinmsus21:/opt/softwareag/CentraSite/utilities> ./CentraSiteCommand.sh
unload auditlog -user Administrator -password manage
```

The response to this command is in XML as follows:

```
Executing the command : unload auditLog
<?xml version="1.0" encoding="UTF-8" ?><ino:response
xmlns:ino="http://namespaces.softwareag.com/tamino/response 2"
xmlns:xql="http://metalab.unc.edu/xql/"><ino:message
ino:returnvalue="0"><ino:messageline>starting admin command
ino:AuditLog(&#39;unload&#39;,&#39;&#39;)</ino:messageline></ino:message>
<AuditLog no='1'>
<A t='1970-01-01T00:00:00Z' a='A' />
<A t='2015-02-17T14:31:33Z' a='S' u='INTERNAL\Administrator' />
<A t='2015-02-17T14:31:41Z' a='0' u='INTERNAL\Administrator' />
<A t='2015-02-17T14:31:41Z' a='U' u='INTERNAL\Administrator' />
</AuditLog>
<ino:message ino:returnvalue="0"><ino:messageline>admin command
ino:AuditLog(&#39;unload&#39;,&#39;&#39;)
completed</ino:messageline></ino:message></ino:response>
Successfully executed the command : unload auditLog
```

Code	Audit Log Type
A	Active
I	Inactive
P	Purge
R	Switch
S	Start
T	Stop
U	unload

Code	Audit Log Type
V	User data
X	Unused
Y	Unknown
Z	Record
0	Connect
1	Create
2	Update
3	Delete
4	Set Access Control List (ACL)
5	Update ACL
6	Unset ACL
7	Commit
8	Rollback
9	Disconnect

Note:
You might also see the following additional attributes in the log entries:

- `f` - File number
- `d` - Doc type (if it is available)
- `i` - ID (INO-ID) of the document added/updated/deleted.
- `n` - Name (if it is available)
- `k` - UDDI key
- `p` - Payload (p=1, if the log entry contains a payload; p=2, if a log entry has a payload information, but it is too large to be included in the log entry.
- `s` - Session number

Stopping the Audit Logs

To stop the logging of CentraSite user information through the Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `stop auditLog` for this purpose.

To stop the logging of CentraSite user information

- Run the command `stop auditLog`.

The syntax is of the format `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd stop auditLog [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD>`

The input parameters are:

Parameter	Description
`CENTRASITE-URL`	(Optional). The URL of the CentraSite registry. For example, `http://localhost:53307/CentraSite/CentraSite`.
`USER-ID`	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, `Administrator`.
`PASSWORD`	The password for the registered CentraSite user.

Example:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd stop auditLog -user
Administrator -password manage -url http://localhost:53307/CentraSite/CentraSite
```

The response to this command could be:

```
Executing the command : stop auditLog

Successfully executed the command : stop auditLog
```

Checking Current State of Audit Log

Pre-requisites:

To check the state of audit logging through the Command Line Interface, you must have the CentraSite Administrator role.

In certain circumstances, you may want to check if audit logging is `On` or `Off`. By default, the audit log is `Off`.

CentraSite provides a command tool named `state auditLog` for this purpose.

To check the state of audit logging

- Run the command `state auditLog`.

  The syntax is of the format `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd state auditLog [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD>`

  The input parameters are:

Parameter	Description
`CENTRASITE-URL`	(Optional). The URL of the CentraSite registry. For example, `http://localhost:53307/CentraSite/CentraSite`.

Parameter	Description
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, Administrator.
PASSWORD	The password for the registered CentraSite user.

Example:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd state auditLog -user
Administrator -password manage -url http://localhost:53307/CentraSite/CentraSite
```

The response to this command could be:

```
Executing the command : state auditLog

Successfully executed the command : state auditLog
```

Switching the Audit Log Store

Pre-requisites:

To change the storage of audit logs through the Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `switch auditLog` for this purpose.

When you use the `start auditLog` command, CentraSite logs the information in the current audit log. You can use the `switch auditLog` command to log information in the next audit log file. For example, if the current log file is audit log 1, you can use the `switch auditLog` command to specify that CentraSite should store the next log entry in the subsequent log file (audit log 2). CentraSite can create up to ten audit log files after you successfully run the `start auditLog` command.

To change the storage of audit logs

- Run the command `switch auditLog`.

  The syntax is of the format `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd switch auditLog [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD>`

  The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, http://localhost:53307/CentraSite/CentraSite.
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, Administrator.
PASSWORD	The password for the registered CentraSite user.

Parameter	Description
LOGDATA	The data to be written to the audit log.

Example (UNIX):

```
/opt/softwareag/CentraSite/utilities> ./CentraSiteCommand.sh switch auditlog -user
Administrator -password manage
```

Recording the Audit Log

Pre-requisites:

To add information into the current audit log through the Command Line Interface, you must have the CentraSite Administrator role.

In certain circumstances, you might want to directly insert information into the current audit log for specific data items. For example, you might want to insert a log entry into the current audit log at the beginning or end of a set of actions.

CentraSite provides a command tool named `record auditLog` for this purpose. The tool takes an additional parameter, `LOGDATA` to indicate the data item to insert in the audit log. The audit log entry corresponding to this command is inserted into the audit log file with the type, `Z`.

To add information into the current audit log

- Run the command `record auditLog`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd record auditLog [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> [-inputData <LOGDATA>]`

  The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, `http://localhost:53307/CentraSite/CentraSite`.
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, `Administrator`.
PASSWORD	The password for the registered CentraSite user.
LOGDATA	The data to be written to the audit log.

  Example (UNIX):

```
/opt/softwareag/CentraSite/utilities> ./CentraSiteCommand.sh record auditlog -user
Administrator -password manage
```

Purging Logs

Pre-requisites:

To configure the log purging, you must have the CentraSite Registry Repository up and running.

For performance reasons, CentraSite uses system log records to store activities and performances of some of the registry objects (say, policies, assets, and so on) that are defined through the log settings. However, in some cases, you might want to modify the log settings configuration to avoid performance degradation of CentraSite.

Note that CentraSite does not purge the following information:

- Policy logs in InProgress state.
- Auditable events that log on object creation.
- Approval logs in Pending state.
- Consumer registration data.
- Purged audit log data (that is, records of the audit logs that were purged earlier).

Note:
When you purge policy log entries, CentraSite creates a new informational policy log entry indicating that policy log entries have been purged. If any such informational policy log entries exist from previous invocations of the policy log purger, they are also purged, so that the only such policy log entry remaining is the one for the current invocation.

Invoking the log purging settings is a two-step process:

1. Create a log purging configuration (`config.xml`) file.
2. Execute the script file from a command line with appropriate input parameters.

Creating Log Purging Configuration File

You create a log purging configuration file, config.xml, to define the log purge settings.

Ensure that the log configuration file contains the XML namespaces for providing uniquely named elements and attributes.

Defining the Log Purge Settings

The `LogPurgeSetting` element specifies the type of log record to be purged from the CentraSite Registry Repository.

In the `LogPurgeSetting` element, add an attribute called `log` and assign it one of the following values:

Log Unit	Description
`Policy_Log`	Purge log records from the Policy Log, which contains information about the design/change-time policies that CentraSite has executed.
`Approval_Log`	Purge log records from the Approval Log, which contains information about all approval requests that have been triggered by a policy with an approval workflow action.
`Audit_Log`	Purge log records from the Audit Log, which contains information on the creation/update activities of a particular asset (including changes in an asset's lifecycle state).
`Runtime_Event_Log`	Purge log records from the Run-Time Event Log, which contains information about run-time events that have occurred in a gateway (a policy-enforcement point (PEP) or a run-time monitoring component).
`Runtime_Performance_Log`	Purge log records from the Run-Time Performance Log, which contains information about run-time metrics for all assets.

This `LogPurgeSetting` element must look similar to this:

```
<LogPurgeSetting log="Runtime_Event_Log">
</LogPurgeSetting>

<LogPurgeSetting log="Runtime_Performance_Log">
</LogPurgeSetting>
```

Define a Log Purge Type

The log purge type attributes provide information about the different type of logs configured to be purged on different days, times, and even how many days of logs to keep.

You can create the purge type attributes that is used by the log purge setting definition.

Within the `LogPurgeSetting` element, create the following attributes (as required) and assign values.

Log Purge Setting	Description
`ExportLocation`	Specifies a location for the exported files on the database.
`Until`	Deletes log records that are older than the specified date/time and then exports the records as an archive to the directory specified on the `ExportLocation` attribute. If there is no `ExportLocation` attribute specified for the log unit, CentraSite deletes the records without exporting them.

Log Purge Setting	Description
	Use xs:dateTime values to indicate the date and time. The default time zone is UTC/GMT.

`OlderThan`	Deletes log records that are older than the specified time interval and then exports the records as an archive to the directory specified on the `ExportLocation` attribute. If there is no `ExportLocation` attribute specified for the log unit, CentraSite deletes the records without exporting them.
	
	Use xs:duration values to indicate the time interval.
	
	For example, to purge all log entries older than 1 month, 10 days, specify:
	
	`<OlderThan>P1M10D</OlderThan>`
	
	To purge all log entries, specify:
	
	`<OlderThan>P0D</OlderThan>`
	
	Note:
	Specify only positive durations or `0`. Negative durations will delete all log entries.

`CommitThreshold`	Defines the threshold value for batch purging each of the log units. The default threshold value is `10000`.

The LogPurgeSetting element must now look similar to this:

```
<LogPurgeSetting log="Runtime_Event_Log">
     <OlderThan>P5D</OlderThan>
</LogPurgeSetting>

<LogPurgeSetting log="Runtime_Performance_Log">
     <Until>2014-05-30</Until>
     <CommitThreshold>1000</CommitThreshold>
</LogPurgeSetting>
```

The final `config.xml` should look similar to this:

```
<LogPurgeSettings
  xmlns="http://namespaces.centrasite.com/configurations/logs">
    <LogPurgeSetting log="Approval_Log">
        <ExportLocation>/home/usr/admin/centrasite/log/backups/approval
        </ExportLocation>
    </LogPurgeSetting>
    <LogPurgeSetting log="Runtime_Event_Log">
        <OlderThan>P5D</OlderThan>
    </LogPurgeSetting>
    <LogPurgeSetting log="Runtime_Performance_Log">
        <Until>2014-05-30</Until>
        <CommitThreshold>1000</CommitThreshold>
    </LogPurgeSetting>
```

```
</LogPurgeSettings>
```

Note:
Ensure that you copy this file somewhere within the file system of the machine where CentraSite is installed.

Purging Logs

Pre-requisites:

To purge logs through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `purge Logs` for this purpose.

To purge logs

- Run the command `purge Logs`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd purge Logs [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> [-assets <AssetName/Key(s)>] -file <CONFIG-FILE>`

  The input parameters are:

Parameter	Description
`CENTRASITE-URL`	(Optional). The URL of the CentraSite registry. Default value is `http://localhost:53307`.
`USER-ID`	The user ID of a registered CentraSite user. For example, a user who has the CentraSite Administrator role.
`PASSWORD`	The password for the registered CentraSite user identified by the parameter `USER-ID`.
`Asset Name/Key (s)`	The name or UUID assigned to the asset and which uniquely identifies it within the registry.
`CONFIG-FILE`	The absolute or relative path to the XML configuration file. If relative, the path should be relative to the location from where the command is executed.

  Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd purge Logs -url "http://localhost:53307/CentraSite/CentraSite" -user "Administrator" -password "manage" -assets "MyService" -file "config.xml"
```

  The response to this command could be:

```
Executing the command : purge Logs
```

```
Successfully executed the command : purge Logs
```

Note:
Changes to the log configuration do not affect the currently running tasks.

Exporting the Purged Log Records

You can selectively export log records to a location on disk.

You can modify the `config.xml` file to specify the export location of purged log records, the date range to purge log records, and execute the command line utility to export the defined log records. For example, if you specify a directory location and the date until which the log records need to be purged, then executing the command line, CentraSite deletes the log records that were generated until the specified date and exports them as archive to the specified directory.

The following table identifies the log purging behavior of CentraSite for each purge setting configuration:

Log Purge Settings	**Until**		**Older Than**	
	Export Location Specified	**Export Location Not Specified**	**Export Location Specified**	**Export Location Not Specified**
Date Specified	Deletes log records that are generated until the specified date and exports them as an archive to the specified directory that can be imported later.	Permanently deletes log records that were generated until the specified date from the CentraSite Registry Repository.	Deletes log records that are older than the specified date and exports them as an archive to the specified directory that can be imported later.	Permanently deletes log records that are older than the specified date from the CentraSite Registry Repository.
Date Not Specified	Deletes log records that are generated until the current date and exports them as an archive to the specified directory that can be imported later.	Permanently deletes log records that are generated until the current date from the CentraSite Registry Repository.	Deletes log records until the current date and exports them as an archive to the specified directory that can be imported later.	Permanently deletes log records until the current date from the CentraSite Registry Repository.

Configuring Purger Properties for High Volume Data Handling

Over time, a very large number of log records accumulates in the CentraSite Registry Repository, which makes handling of log records difficult and directly affects the performance of a Registry Repository. Therefore, it is important to purge unwanted log records from the CentraSite Registry Repository occasionally.

However, purging large number of log records at one time would definitely result in system failures. In order to avoid such failures, CentraSite supports purging of the log records in a batch mode. Purging of log records in a batch mode can be initiated by enabling the `enable.partial.commit` property to `true` in the `purger.properties` file. The `purger.properties` file is located in the `logpurging/resources` folder under the CentraSite installation directory.

Note:
Disabling the `enable.partial.commit=true` property would lead to failures if the system is not able to process the huge amount of log records.

Basically, the log records accumulate at a different rate for each log unit. CentraSite defines a default threshold value for batch purging each of the log units as follows:

```
##partial commit enabled/disabled
enable.partial.commit=true
##policy log threshold
policy.log.partial.commit.threshold=1000
##approval log threshold
approval.log.partial.commit.threshold=500
##events threshold
runtime.events.partial.commit.threshold=1000
##metrics log threshold
runtime.performance.log.partial.commit.threshold=1000
## No of days before the current day for which the data does not be included
## in the purging
## in case of Export Log Entries & Delete Log Entries
## (i.e. No date criteria specified).
## purge.older.than=1
```

You can modify the threshold values for batch purging the log units as required.

Note:
If the purging type in the `config.xml` file is set to **Export** (No date criteria), then purging would exclude the log records by the number of days defined in the `purge.older.than` property. For example, if the purging is scheduled with `purge.older.than=7`, then log records that are available ahead of the 7 days would be purged from the CentraSite Registry Repository.

To change the batch purging properties

1. Open the `purger.properties` file in a rich text editor. You can find the file in the *`Software AG_directory`*`/CentraSite/logpurging/resources` directory.

2. Set the batch purging property, `enable.partial.commit`, to `true` or `false`, as required. The default value is `true`.

3. Specify the threshold values in the `partial.commit.threshold` property for a log unit.

4. Save and close the file.

   The changes take effect in the next purging.

Removing Leftover Auditable Events

Pre-requisites:

To remove the leftover auditable events through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

In previous releases of CentraSite, some auditable events remained in the log after purging, even though the events referred to CentraSite objects that no longer existed.

There is one known situation where many of these events were created: the Integration Server can be configured to write metrics information for Virtual Services into CentraSite at regular intervals. Each metrics update resulted in some leftover events.

If you have upgraded your CentraSite Registry Repository from a previous release, such leftover auditable events might still exist in the log.

CentraSite provides a Java tool named `CentraSiteOptimizeAuditableEvents.jar` for this purpose.

To remove the leftover auditable events from the command line

- Run the Java tool `CentraSiteOptimizeAuditableEvents.jar`.

  The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd CentraSiteOptimizeAuditableEvents.jar <CentraSite URL> <admin user id> <password>`

  The input parameters are:

Parameter	Description
`CentraSite URL`	The URL of the CentraSite registry. For example, `http://localhost:53307/CentraSite/CentraSite`.
`admin user id`	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, `Administrator`.
`password`	The password for the registered CentraSite user identified by the parameter `admin user id`.

  Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd
CentraSiteOptimizeAuditableEvents.jar "http://localhost:53307/CentraSite/CentraSite"
DOMAIN\admin pAsSw0rD
```

3 Administering CentraSite with Command Central

- Overview 124
- Viewing CentraSite Components 124
- Changing the Authentication Mode 125
- Changing the Administrator User Password for Managed Products 125
- Verifying the Outbound Authentication Settings 125
- Commands that CentraSite Application Server Tier Supports 126
- Commands that CentraSite Registry Repository Supports 126
- Lifecycle Actions for CentraSite Registry Repository 127
- Run-time Monitoring Statuses for CentraSite Registry Repository 127

Overview

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.
- 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 the Command Central help.

Viewing CentraSite Components

In certain circumstances, you might want to view the CentraSite Registry Repository (CRR) and CentraSite Application Server Tier (CAST).

The CentraSite Registry Repository is a process while the 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 **CentraSite Registry 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 **CentraSite Application Server**.

Changing the Authentication Mode

In the instance **Overview** tab, click  in the **Authentication** field to change the authentication mode using the **Authentication Mode** dialog box.

By default, Command Central uses basic authentication with a **Fixed User** to communicate with Platform Manager. With **Fixed User** authentication, the authentication credentials (Administrator / manage) for the Platform Manager is fixed.

Changing the Administrator User Password for Managed Products

You change the Administrator password for CTP 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 CTP.

3. On the Configuration tab, select **Users**.

4. On the Users page, click **Administrator**.

5. Click **Edit** and specify the new password for **Administrator**.

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.

Commands that CentraSite Application Server Tier Supports

The CentraSite Application Server Tier (CAST) runtime component supports the Platform Manager command listed in the following table. The table lists where you can find general information about the command.

Commands	Description
`sagcc get inventory components node_alias INM2AplSrv-ENGINE [options ]`	Retrieves information about CAST runtime component.

For information on usage of this command, see the Command Central help.

Note:
The CentraSite Application Server Tier does not support the `sagcc 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).

Commands that CentraSite Registry Repository Supports

The CentraSite Registry Repository runtime 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 runtime component supports or details about the actions it takes when you execute an `exec` command.

Commands	For more information, see...
`sagcc get inventory components node_alias INM2RegRep-PROCESS [options ]`	Retrieves information about CRR runtime component.
`sagcc exec lifecycle action node_alias INM2RegRep-PROCESS [options ]`	Executes an action against CRR runtime components. For CentraSite-specific information about supported actions, see "Lifecycle Actions for CentraSite Registry Repository" on page 127.
`sagcc get monitoring state runtimeComponentId=INM2RegRep-PROCESS [nodeAlias=alias ]`	Retrieves the runtime status and runtime state of a runtime component.

Commands	For more information, see...
`[includeChildren=true] [refresh=true] [options ]`	Runtime status indicates whether a runtime component is running or not. Runtime state indicates the health of a runtime component by providing (Key Performance Indicators (KPIs) for the component. For CentraSite-specific runtime statuses, see "Run-time Monitoring Statuses for CentraSite Registry Repository" on page 127.

For information on usage of these commands, see the Command Central help.

Lifecycle Actions for CentraSite Registry Repository

The following table lists the actions that the CentraSite Registry Repository (CRR) run-time component supports with the `sagcc exec lifecycle` command and the operation taken against the run-time component when an action is executed.

Action	Description
`start`	Starts 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. **Note:** The CentraSite Server waits for currently active processing to finish before stopping.
`restart`	Stops, then restarts the run-time component. The run-time status is set to ONLINE.
`startindebugmode`	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 *CentraSite_directory* \data directory.

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 `sagcc get monitoring runtimestatus` and `sagcc 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.

4 CentraSite Command Line Tools

- Introduction to CentraSite Command Line Tools 130
- INOADMIN Command Line Tools 130
- CentraSiteCommand Line Tools 135
- CentraSiteToolbox Command Line Tools 145
- Configuring CentraSiteCommand to Use SSL 147

Introduction to CentraSite Command Line Tools

CentraSite provides several command line tools that you can use to retrieve CentraSite Registry Repository (CRR) information and perform administrative tasks in a shell-oriented environment.

The CentraSite Command Line (CLI) provides a variety of commands classified into the following categories:

- CentraSite INOADMIN Commands, called as `inoadmin.[cmd|sh]`
- CentraSite Commands, called as `CentraSiteCommand.[cmd|sh]`
- CentraSite Toolbox Commands, called as `CentraSiteToolbox.[cmd|sh]`

INOADMIN Command Line Tools

The `inoadmin` command line tool offers functionality for performing low-level administrative operations on the CentraSite Registry Repository. The functionality provided by `inoadmin` command tool includes:

- Starting and stopping the CentraSite Registry Repository.
- Maintaining the internal database that houses the CentraSite Registry Repository.
- Configuring port numbers used by the CentraSite Registry Repository.
- Configuring secure communication between the CentraSite Registry Repository and the CentraSite Application Server Tier (CAST).

You execute the `inoadmin` command in the command line tool `inoadmin.cmd` (Windows) or `inoadmin.sh` (UNIX). The command line tool is located in the `utilities` folder in the *<CentraSiteInstall_Directory>*. If you call the `inoadmin` command without parameters, you receive a summary of the command syntax and the available functions.

You can use environment variables to configure the `inoadmin` command tool behavior:

Environment variable	Purpose
`INOADMIN_OUTPUT_TXT`	Determines whether the `inoadmin` command output is displayed as an XML document or as a formatted table. If the value is null (that is, if the value is not defined), the `inoadmin` output is displayed as an XML document. If any non-null value is specified, a formatted table is displayed.
`INOADMIN_NO_ MESSAGE_OUTPUT`	Determines whether progress status messages are output while `inoadmin` calls are running. If this variable is null (that is, if the value is not defined), all progress status messages are output. If any other value is set for this variable, the progress status messages are suppressed and only the result is displayed.

Environment variable	Purpose
INOADMIN_RUN_AS_JOB	Creates a job log that contains status information generated while inoadmin is running.

If you use inoadmin within a command script, you can use the return status to verify the successful completion of the command. A zero return status means that the command executed successfully, whereas a non-zero return status means that the command did not execute successfully.

The syntax is of the format:

```
inoadmin <inoadminScript> <inoadminScript Parameter>
```

Example:

```
inoadmin setlocation CentraSite <LocationType> <path> [<path> ...]
```

List of inoadmin Command Line Tools

Command Name	Description
setlocation	Modifies the currently defined path of a location. For more information, see "Configuring CentraSite Database Locations" on page 31.
showactivity	Displays the database activity information. For more information, see "Displaying the Database Activity" on page 22.
showlocations	Shows a list of all the currently defined locations. For more information, see "Configuring CentraSite Database Locations" on page 31.
allowreplication	Registers the host name of the replication instance. For more information, see "Setting Up Replication Instances of Registry Repository" on page 36.
denyreplication	Deregisters the host name of the replication instance to prevent further re-replication. For more information, see "Setting Up Replication Instances of Registry Repository" on page 36.
setreplication	Sets a copy of the CentraSite Registry Repository as replication instance of the master on the given host. For more information, see "Setting Up Replication Instances of Registry Repository" on page 36.

Command Name	Description
`resetreplication`	Resets the replication instance to a normal server instance to use instead of the master instance which is lost for any reason. For more information, see "Setting Up Replication Instances of Registry Repository" on page 36.
`replicationinfo`	Indicates if the server is a master server or a replication server. For more information, see "Setting Up Replication Instances of Registry Repository" on page 36.
`listproperties`	Shows a list of all of the available database properties and their values. For more information, see "Database Configuration Parameters" on page 33.
`getproperty`	Shows the value of a given property. For more information, see "Database Configuration Parameters" on page 33.
`setproperty`	Modifies the value of a given property with the new value. For more information, see "Database Configuration Parameters" on page 33.
`setproperties`	Modifies the values of several properties with the values specified in the (supplied) XML file. For more information, see "Database Configuration Parameters" on page 33.
`start CentraSite`	Starts the CentraSite Registry Repository. For more information, see "Starting and Stopping the Application Server Tier and Registry Repository" on page 11.
`stop CentraSite`	Terminates the CentraSite Registry Repository session normally and waits for currently active user transactions to finish. For more information, see "Starting and Stopping the Application Server Tier and Registry Repository" on page 11.

Command Name	Description
`stop CentraSite normal`	Terminates the CentraSite Registry Repository session normally. For more information, see "Starting and Stopping the Application Server Tier and Registry Repository" on page 11.
`stop CentraSite rollback`	Terminates the CentraSite Registry Repository immediately and rollbacks the currently active user transactions. For more information, see "Starting and Stopping the Application Server Tier and Registry Repository" on page 11.
`stop CentraSite abort`	Causes an emergency shutdown of the CentraSite Registry Repository. For more information, see "Starting and Stopping the Application Server Tier and Registry Repository" on page 11.
`backup`	Copies the contents of CentraSite internal database on the file system For more information, see "Backing Up the Database" on page 26.
`listbackups`	Shows a list of all available backups and their backup keys. For more information, see "Displaying Backup List" on page 25.
`deletebackup`	Deletes a backup that is no longer required. For more information, see "Deleting a Backup" on page 29.
`restore`	Restores the contents of CentraSite 's internal database back to a previous state. For more information, see the "Restoring the Database from a Backup" on page 27 *CentraSite Administrator's Guide*.
`reorganize`	Reclaims disk space in the CentraSite internal database. For more information, see "Reclaiming Disk Space in CentraSite Database" on page 35.

Command Name	Description
`listdbspaces`	Displays information about the physical disk requirements of CentraSite 's internal database. For more information, see "Displaying the Database Space" on page 24.

Return Codes from inoadmin Command Execution

The following table describes the return codes you might encounter when using the command line tool `inoadmin`.

Return Code	Description
0	Execution of the command was successful.
1	A required parameter was not specified.
2	The command includes an invalid parameter.
3	The command includes an invalid number of parameters.
4	The output that a command returned does not match the expected values specified with the `version` option.
5	The server was not registered.
6	The server already exists.
7	The state of the server is active.
8	The state of the server is inactive.
9	The server startup failed.
10	The specified file cannot be located. Make sure you have entered the correct path and file name.
11	A parameter specified an invalid name.
12	A parameter specified an invalid value.
13	An error occurred during command execution.
14	Access to the operating system file is denied.
15	The backup deletion failed.
16	The specified service cannot be found.
17	First time startup of the service failed.
18	First time startup of the service failed.

Return Code	Description
19	The directory name is not valid.
20	Indicates that the dbspace name is invalid.
21	An autorepair is pending.
22	An operating system file is locked by the server.
23	No space left on the database.
24	There is not enough memory to run the script.
25	The server to be set as replication instance was not created from a backup of the specified master.
26	The command is not allowed for a read-only server.
27	Database already updated cannot be set up as replication instance.
28	Master server must be restarted to set up a replication instance.
29	The command is not allowed for a master server.
30	The command is not allowed for a replication server.
31	The specified entry already exists.
32	The specified entry was not found.
33	The command is not allowed for a former replication database.

CentraSiteCommand Line Tools

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

- Infrastructure engineers can see at a glance which products and fixes are installed where, and can easily compare installations to find discrepancies.
- System administrators can configure environments using a graphical user interface, command line tool, or API, so maintenance can be performed with 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.

You execute the `CentraSiteCommand` in the command line tool `CentraSiteCommand.cmd` (Windows) or `CentraSiteCommand.sh` (UNIX) to perform various administrative tasks in CentraSite. The command line tool is located in the `utilities` folder in the *<CentraSiteInstall_Directory>*.

The syntax is of the format:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.[cmd|sh] <CentraSiteCommand>
<CentraSiteCommand Parameter>
```

If you start the command tool with no parameters, you receive a help text summarizing the required input parameters.

The parameters of the command are case-sensitive, so for example the parameter `-url` must be specified as shown and not as `-URL`.

Example:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set preferences -user <USER-ID>
-password <PASSWORD> [-url <CENTRASITE-URL>] -targetUser <TARGET USER> -file <CONFIGFILE>
```

List of CentraSiteCommand Line Tools

Command Name	Description
`set DefaultDomain`	Sets a default authentication configuration in CentraSite. For more information, see "Creating and Maintaining Authentication Configurations" on page 68.
`get Authentication`	Fetches information of an existing LDAP authentication configuration in CentraSite. For more information, see "Creating and Maintaining Authentication Configurations" on page 68.
`set Authentication`	Adds a new (LDAP) authentication configuration to CentraSite. For more information, see "Creating and Maintaining Authentication Configurations" on page 68.
`remove Authentication`	Deletes an existing LDAP authentication configuration in CentraSite. For more information, see "Creating and Maintaining Authentication Configurations" on page 68.

Command Name	Description
`list gateways`	Shows a list of all currently defined gateways in CentraSite. For more information, see *CentraSite User's Guide*.
`set gateway`	Adds a new Mediator gateway to CentraSite. For more information, see *CentraSite User's Guide*.
`remove gateway`	Deletes an existing Mediator gateway in CentraSite. For more information, see *CentraSite User's Guide*.
`state runtimeEvent`	Fetches the current state of an active run-time event store in CentraSite. For more information, see *CentraSite User's Guide*.
`switch runtimeEvent`	Changes the storage mode of run-time events in CentraSite. For more information, see *CentraSite User's Guide*.
`purge runtimeEvent`	Removes unwanted data of the active run-time event store in CentraSite. For more information, see *CentraSite User's Guide*.
`set SSL RR`	Specifies the SSL security values for CentraSite Registry Repository (CRR). For more information, see "Setting Security Configuration for CentraSite Registry Repository" on page 42.
`set SSL AST`	Specifies the SSL security values for CentraSite Application Server Tier (CAST). For more information, see "Setting Security Configuration for CentraSite Application Server Tier Components" on page 44.
`set SSL usage`	Changes the communication between CentraSite Application Server Tier and CentraSite Registry Repository from a full 2-way SSL (HTTPS) communication to HTTP communication. For more information, see "Allowing HTTP Communication Between CAST and CRR " on page 46.

Command Name	Description
`get SSL RR`	Fetches the SSL security values of CentraSite Registry Repository. For more information, see "Obtaining Security Configuration of CentraSite Registry Repository" on page 40.
`get SSL AST`	Fetches the SSL security values of CentraSite Application Server Tier. For more information, see "Obtaining Security Configuration of CentraSite Application Server Tier " on page 43.
`get SSL usage`	Identifies the communication method between the CentraSite Application Server Tier (CAST) and the CentraSite Registry Repository (CRR). For more information, see "Identifying the Communication Method Between CAST and CRR" on page 46.
`set Email`	Configures the email notification settings for Simple Mail Transport Protocol (SMTP) server. For more information, see "Configuring Email Server Settings" on page 90.
`get Email`	Fetches the email notification settings of Simple Mail Transport Protocol server. For more information, see "Fetching Email Server Settings" on page 92.
`set preferences`	Defines the user-specific settings for CentraSite Business UI. For more information, see *CentraSite User's Guide*.
`get preferences`	Fetches the user-specific settings for CentraSite Business UI. For more information, see *CentraSite User's Guide*.
`reset preferences`	Resets the user-specific settings in CentraSite Business UI. For more information, see *CentraSite User's Guide*.
`list Reports`	Shows a list of all the existing reports in CentraSite Business UI.

Command Name	Description
	For more information, see *CentraSite User's Guide*.
`add Report`	Adds a new report to the CentraSite.
	For more information, see *CentraSite User's Guide*.
`update Report`	Modifies the information of an existing report in the CentraSite Registry Repository.
	For more information, see *CentraSite User's Guide*.
`delete Report`	Deletes an existing report in CentraSite.
	For more information, see *CentraSite User's Guide*.
`share Report`	Shares dynamic, read-only reports with the API Portal users (without giving them access to all of the report's data) for monitoring API usage.
	For more information, see *CentraSite User's Guide*.
`download Report Template`	Downloads a BIRT report design file that contains the definition of a report template.
	For more information, see *CentraSite User's Guide*.
`list Scheduled Reports`	Shows a list of all the existing scheduled reports in CentraSite.
	For more information, see *CentraSite User's Guide*.
`add Scheduled Report`	Adds a new scheduled report to CentraSite.
	For more information, see *CentraSite User's Guide*.
`update Scheduled Report`	Modifies the information of an existing scheduled report in the CentraSite Registry Repository.
	For more information, see *CentraSite User's Guide*.
`delete Scheduled Report`	Deletes an existing scheduled report in CentraSite.
	For more information, see *CentraSite User's Guide*.
`trigger Scheduled Report`	Triggers an existing scheduled report in CentraSite.
	For more information, see *CentraSite User's Guide*.
`list Pending Consumer Registrations`	Shows a list of all the pending consumer registration requests that were pending for an approval and were transferred from a previous version of CentraSite in the migration process.

Command Name	Description
	For more information, see *CentraSite User's Guide*.
`add Admin`	Adds a user with the CentraSite Administrator role in CentraSite's user repository. For more information, see *CentraSite User's Guide*.
`generate JaasConfiguration`	Transforms and migrates the LDAP configuration to the new JAAS configuration. For more information, see "Transforming and Migrating Internal and LDAP Configuration Data" on page 86.
`restore accesstokens`	Restores the expired access tokens in CentraSite. For more information, see *CentraSite User's Guide*.
`deploy`	Deploys (also referred to as Publish in CentraSite Business UI) a single Virtual Service to Mediator gateway. For more information, see *CentraSite User's Guide*.
`undeploy`	Undeploys (also referred to as Unpublish in CentraSite Business UI) a single Virtual Service from Mediator gateway. For more information, see *CentraSite User's Guide*.
`bulk deploy`	Deploys (also referred to as Publish in CentraSite Business UI) multiple Virtual Services to Mediator gateways in one line. For more information, see *CentraSite User's Guide*.
`bulk undeploy`	Undeploys (also referred to as Unpublish in CentraSite Business UI) multiple Virtual Services from Mediator gateway in one line. For more information, see *CentraSite User's Guide*.
`bulk redeploy`	Redeploys (also referred to as Publish in CentraSite Business UI) multiple Virtual Services to a Mediator gateway in one line. For more information, see *CentraSite User's Guide*.
`bulk clean redeploy`	Undeploys and redeploys multiple Virtual Services that are already deployed to a Mediator gateway in one line.

Command Name	Description
	For more information, see *CentraSite User's Guide*.
`delete Organization`	Deletes an organization that includes internal references to other registry objects in the CentraSite Registry Repository. For more information, see *CentraSite User's Guide*.
`add Report Association`	Associates a report with the specified asset types. For more information, see *CentraSite User's Guide*.
`list Report Associations`	Shows a list of all the asset types that are associated with a report. For more information, see *CentraSite User's Guide*.
`delete Report Association`	Deletes the association between a report and the asset types. For more information, see *CentraSite User's Guide*.
`set Password`	Changes the password of a predefined user or a login user. For more information, see *CentraSite User's Guide*.
`import Service`	Imports a REST Service asset using RAML (RESTful API Modeling Language) or Swagger file into the CentraSite Registry Repository. For more information, see *CentraSite User's Guide*.
`ImportWSDL`	Imports a Web Service asset using WSDL (Web Services Description Language) file into the CentraSite Registry Repository. For more information, see *CentraSite User's Guide*.
`ImportSchema`	Imports an XML Schema asset using XML Schema Definition (XSD) file into the CentraSite Registry Repository. For more information, see *CentraSite User's Guide*.
`ImportBPEL`	Imports a BPEL Process asset using BPEL file into the CentraSite Registry Repository. For more information, see Importing a BPEL Process. For more information, see *CentraSite User's Guide*.

Command Name	Description
`ImportXPDL`	Imports a Process asset using XPDL file into the CentraSite Registry Repository. For more information, see *CentraSite User's Guide*.
`ImportArchive`	Imports assets using an archive (.zip) file into the CentraSite Registry Repository. For more information, see *CentraSite User's Guide*.
`start auditLog`	Starts logging of the CentraSite user information. For more information, see "Starting the Audit Logs" on page 108.
`unload auditLog`	Reads the audit log and displays the information as a standard output or writes the information to a file. For more information, see "Storing the Audit Log Data" on page 109.
`stop auditLog`	Stops logging of the CentraSite user information. For more information, see "Stopping the Audit Logs" on page 111.
`state auditLog`	Checks if the user information was successfully logged on or not in the CentraSite Registry Repository. For more information, see "Checking Current State of Audit Log" on page 112.
`switch auditLog`	Specifies that CentraSite should store the next log entry in the subsequent log file. For more information, see "Switching the Audit Log Store" on page 113.
`record auditLog`	Inserts a log entry into the current audit log. For more information, see "Recording the Audit Log" on page 114.
`set Log`	Defines the log settings for specified log units using a configuration file. For more information, see "Configuring Log Settings" on page 103.
`get Log`	Fetches the configuration details of the log settings.

Command Name	Description
	For more information, see "Obtaining Log Configuration Settings" on page 107.
`purge Logs`	Purges the log configuration settings. For more information, see "Purging Logs" on page 118.
`sync consumers`	Synchronizes consumer applications with a Mediator gateway. For more information, see *CentraSite User's Guide*.
`get Asymmetric Binding`	Fetches information of the current asymmetric binding configuration in CentraSite. For more information, see *CentraSite User's Guide*.
`set Asymmetric Binding`	Specifies an asymmetric binding configuration in CentraSite. For more information, see *CentraSite User's Guide*.
`remove Asymmetric Binding`	Removes an existing asymmetric binding configuration from CentraSite. For more information, see *CentraSite User's Guide*.
`set UDDI`	Specifies the local UDDI properties or global UDDI properties in CentraSite.
`get UDDI`	Fetches information of the local UDDI properties or global UDDI properties in CentraSite.
`purge accesstokens`	Purges all the expired or inactive access tokens in CentraSite. For more information, see *CentraSite User's Guide*.
`remove Attribute`	Removes an attribute from an asset type, in cases where there are existing asset instances of the asset type containing a value for the attribute to be removed from the CentraSite Registry Repository. For more information, see *CentraSite User's Guide*.
`add Search`	Adds a custom XQuery reporting search to the CentraSite Registry Repository. For more information, see *CentraSite User's Guide*.

Command Name	Description
`copy Search`	Creates a copy of an XQuery reporting search in the CentraSite Registry Repository. For more information, see *CentraSite User's Guide*.
`get Search`	Retrieves information about an XQuery reporting search in the CentraSite Registry Repository. For more information, see *CentraSite User's Guide*.
`list Search`	Shows a list of all XQuery reporting searches in CentraSite. For more information, see *CentraSite User's Guide*.
`get ApiPortalConfig`	Fetches information of the current retry configuration settings for delivery of messages from CentraSite to API Portal. For more information, see *CentraSite User's Guide*.
`set ApiPortalConfig`	Specifies the retry configuration settings for delivery of messages from CentraSite to API Portal. For more information, see *CentraSite User's Guide*.
`send AccessTokens`	Resends the undelivered access tokens to the respective API Portal gateways. For more information, see *CentraSite User's Guide*.
`list CAST`	Shows a list of available CASTs in CentraSite.
`add CAST`	Adds a new CAST to the CentraSite Registry Repository.
`remove CAST`	Deletes an existing CAST from the CentraSite Registry Repository.
`set Consumer`	Associates an Application asset (represented as a consumer application) with a Virtual Service asset. For more information, see *CentraSite User's Guide*.
`reset Consumer`	Removes an association that exists between an Application asset (represented as a consumer application) and a Virtual Service asset. For more information, see *CentraSite User's Guide*.

Command Name	Description
`reassign Consumer`	Reassigns an Application, API-Key, or OAuth2 Client asset (represented as a consumer application) to a Virtual Service asset. For more information, see *CentraSite User's Guide*.

CentraSiteToolbox Command Line Tools

The toolbox script assists in running the command line utilities in CentraSite's toolbox. This toolbox is a collection of useful Java Archive (.jar) files. Due to possible classpath issues it might not always be possible to run these Java Archive files on location using `java -jar`. If encountering such issues use the toolbox script `CentraSiteToolbox.[cmd|sh]`.

All necessary classpath settings are performed by the `CentraSiteToolbox` script.

You execute the Java tool in the command line tool `CentraSiteToolbox.cmd` (Windows) or `CentraSiteToolbox.sh` (UNIX) with the Java tool name as first parameter. The command line tool is located in the `utilities` folder and the JAR file is located in the `bin` folder in the *<CentraSiteInstall_Directory>*.

The syntax is of the format:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.[cmd|sh] <ToolboxSript>
<ToolboxScript Parameter>
```

If you start this command line tool with no parameters, you receive a help text summarizing the required input parameters.

The parameters of the Java tool are case-sensitive.

Example:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd CentraSiteCacheConfiguration.jar
http://localhost:53307/CentraSite/CentraSite Administrator manage DISPLAY
```

List of CentraSiteToolbox Command Line Tools

Command Name	Description
`AssetManager`	Change type of an existing asset or a list of assets (saved search) to a specified user-defined virtual asset type or back to the specified base type. For more information, see *CentraSite User's Guide*.
`AssetTypeManager`	Create asset type or delete existing asset type in CentraSite. For more information, see *CentraSite User's Guide*.

Command Name	Description
`CentraSiteAssetImporter`	Import data from Excel sheet or the csv file. The tool will only support .xlsx (Excel sheet) files. For more information, see *CentraSite User's Guide*.
`CentraSiteCacheConfiguration`	Display JAXR-based configuration settings. In addition, the tool allows to modify the existing JAXR-based configuration settings. For more information, see "Configuring Registry Cache Settings" on page 62.
`CentraSiteDeleteAsset`	Delete asset that has internal associations in the CentraSite Registry Repository. For more information, see *CentraSite User's Guide*.
`CentraSiteDeleteAllAsset`	Deletes all the assets in the CentraSite Registry Repository. For more information, see *CentraSite User's Guide*.
`CentraSiteDeleteUser`	Delete a user which has internal references in the CentraSite Registry Repository. For more information, see *CentraSite User's Guide*.
`CentraSiteOptimizeAuditableEvents`	Remove auditable events that remained in the CentraSite Registry Repository after the purging operation. For more information, see *CentraSite User's Guide*.
`EstablishRestrictedLocale`	Restrict users from changing their locale settings on the user preferences (in CentraSite Business UI) or user profile (in CentraSite Control). In addition, the tool allows to set the restricted locale as login locale for all registered users available in the CentraSite registry.
`FixProfileSequenceNumber`	Resolve sequence number in the display order of predefined and custom profiles in the Asset Details page. For more information, see *CentraSite User's Guide*.
`PurgeConsumerRegistrationRequests`	Purge consumer registration requests with a status Approved or Rejected. The tool allows you to purge all consumer registration requests that are related to a specified object or all consumer registration requests that are available in the CentraSite registry.

Command Name	Description
	For more information, see *CentraSite User's Guide*.
`I18NMessageFinder`	Fetch message keys for message texts in the CentraSite Message Database. For more information, see *CentraSite Developer's Guide*.
`PurgePolicyParameters`	Purge orphaned policy parameters that are available in the CentraSite registry. For more information, see *CentraSite User's Guide*.
`LoadModule`	Upload XQuery modules for CentraSite reports. For more information, see *CentraSite Developer's Guide*.
`PurgeRunTimeData`	Purge runtime metrics and events for a specified Virtual Service asset using the name or UDDI key.
`ReassociateGroups`	Reassociate CentraSite group with a new external group ID. For more information, see *CentraSite User's Guide*.
`ReassociateUsers`	Reassociate CentraSite user with a new external user ID. For more information, see *CentraSite User's Guide*.
`UpdatePolicyActionImplementation`	Update implementation file of a policy action template which is currently used by one or more policies. For more information, see *CentraSite User's Guide*.

Configuring CentraSiteCommand to Use SSL

Pre-requisites:

To configure CentraSiteCommand tool to use SSL, you must have the CentraSite Administrator role.

You can configure CentraSiteCommand tool to communicate with CentraSite Application Server Tier through secure HTTP (HTTPS) connection. When configuring CentraSiteCommand tool to use HTTPS with Secure Sockets Layer (SSL), the necessary keystores and truststores must be set in order to access the required security databases and certificates. The truststore should contain a certificate that is accepted by the Tomcat. A list of certificates that are accepted by Software AG Runtime is defined by the keystore entries in the `<SoftwareAG_directory>/profiles/CTP/configuration/com.softwareag.platform.config.propsloader/com.soag.catalina.connector.https.pid-CentraSite.properties` file.

To configure CentraSiteCommand to use SSL

1. Using KeyStore Explorer, open the Tomcat keystore. The default keystore is `localhost_dont_use_in_production.jks` located at `<Software AG_directory>/profiles/CTP/configuration/tomcat/conf/localhost_dont_use_in_production.jks`.

   When prompted for a password, use the default password (`change_this_password`) for keystores.

2. Right-click on the keypair, and choose **Export** > **Export Certificate Chain**.

3. Export the full certificate chain into the `Software AG_directory`:

   ```
   <Software AG_directory>/profiles/CTP/configuration/tomcat/conf/centrasite.cer
   ```

4. Create a keystore in JKS format to use for SSL.

5. Choose **Tools** > **Import Trusted Certificate** to add the exported certificate to the keystore.

6. Select and import the self-signed certificate (`centrasite.cer`) into the new keystore (`centrasitekeystore.jks`).

7. Create a keystore alias using the keystore (`centrasitekeystore.jks`).

   When asked for an alias provide the host name of the machine where CentraSite is installed.

   For more information on creating a keystore alias, see *webMethods Integration Server Administrator's Guide*.

8. Save the new keystore alias as`centrasitekeystore.jks` into the CentraSiteCommand directory:

   ```
   <Software AG_directory>/CentraSite/utilities
   ```

   When prompted for a password to save the keystore alias, the default password for keystores and truststores is `change_this_password`.

9. Open the `CentraSiteCommand.cmd` file in a text editor. You can find the `CentraSiteCommand.cmd` file in the following location:

   ```
   <Software AG_directory>/CentraSite/utilities
   ```

10. Locate the parameter setting `SET FINAL_CMD=%CS_JAVA_EXE%`.

    CentraSiteCommand

    ```
    SET FINAL_CMD=%CS_JAVA_EXE% -Dinstallpath=%CENTRASITE_HOME%
    -DextensionCommand=%CENTRASITE_HOME%\utilities\ExtensionCommand.xml
    -Dlog4j.configuration=file:%CENTRASITE_HOME%\utilities\log4j.xml
    -Djava.util.logging.config.file=%CENTRASITE_HOME%\utilities\logging.properties -cp
    %LOCAL_CLASSPATH% com.softwareag.centrasite.administration.cli.CentrasiteCommand %*
    ```

ToolboxCommand

```
set EXECUTABLE=%CS_JAVA_EXE% -Dcentrasite.installation.root=%CENTRASITE_HOME%
-Dlog4j.configurationFile=log4j2.properties -Djava.util.logging.config.file=unused
-cp %LOCAL_CLASSPATH% %CLASS% %*
```

11. Add the SSL parameter setting:

    CentraSiteCommand

    ```
    SET FINAL_CMD=%CS_JAVA_EXE% -Dinstallpath=%CENTRASITE_HOME%
    -DextensionCommand=%CENTRASITE_HOME%\utilities\ExtensionCommand.xml
    -Dlog4j.configuration=file:%CENTRASITE_HOME%\utilities\log4j.xml
    -Djava.util.logging.config.file=%CENTRASITE_HOME%\utilities\logging.properties
    -Djavax.net.ssl.trustStore=centrasitekeystore.jks
    -Djavax.net.ssl.trustStorePassword=change_this_password
    -Djavax.net.ssl.trustStoreType=jks -cp %LOCAL_CLASSPATH%
    com.softwareag.centrasite.administration.cli.CentrasiteCommand %*
    ```

    ToolboxCommand

    ```
    set EXECUTABLE=%CS_JAVA_EXE% -Dcentrasite.installation.root=%CENTRASITE_HOME%
    -Dlog4j.configurationFile=log4j2.properties -Djava.util.logging.config.file=unused
    -Djavax.net.ssl.trustStore=centrasitekeystore.jks
    -Djavax.net.ssl.trustStorePassword=change_this_password
    -Djavax.net.ssl.trustStoreType=jks -cp %LOCAL_CLASSPATH% %CLASS% %*
    ```

12. Save and close the file. CentraSiteCommand should now be using SSL.