CentraSite

Basic Operations

Version 9.6

April 2014

This document applies to CentraSite Version 9.6.

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

Document ID: IINM-AG-SMH-96-20140318

Table of Contents

Basic Operations v
1 Administering the License Key 1
The Relationship Between the License Key and CentraSite Editions 2
Changing the License Key 3
Working with Time-Limited Licenses 3
2 Maintaining CentraSite's Internal Database 5
Repository Monitoring 6
Backing up the Database 8
Restoring the Database from a Backup 9
Deleting a Backup 11
Moving a CentraSite Database to Another Location 12
Locations 13
Database Configuration Parameters 15
Database Reorganization 16
3 Configuring the Authentication Settings 19
Listing details of a particular configuration 20
Setting the default configuration 23
Adding a configuration 23
Modifying a configuration 26
Removing a configuration 27
Validating a configuration 28
4 Configuring Port Numbers 29
Changing the Port Numbers of the CentraSite Registry/Repository 30
Changing the Software AG Runtime Port Numbers 31
5 Configuring Secure Communication between CentraSite Components 33
Secure Communication between the CRR and the CAST 34
Secure Communication between Software AG Runtime and External Clients 39
6 Configuring the Registry Cache Settings 41
Prerequisites 42
Displaying The Cache Configuration 42
Modifying The Cache Configuration 43
7 Configuring the Email Server 45
Configuring the Email Server Settings 46
Getting the Email Server Settings 48
8 Overview of the CentraSite Administration Tools 51
Overview of the Command Line Tool "inoadmin" 52
Overview of Command Central 53
Return Codes from Command Execution 53

Basic Operations

The CentraSite metadata repository consists of the following components:

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

This document describes how to perform administration-level CentraSite operations, such as the administration of runtime components of CentraSite, administration of the internal database that hosts the registry and repository, performance tuning and error analysis.

Several of the sections in this document refer to the command line tool "inoadmin". For some operations, you can also use features of Command Central. For usage information, see the section **Overview of the CentraSite Administration Tools**.

The documentation consists of the following sections:

Administering the License Key

Maintaining CentraSite's Internal Database

Configuring the Authentication Settings

Configuring Port Numbers

Configuring Secure Communication between CentraSite Components

Configuring the Registry Cache Settings

Configuring the Email Server

Overview of the CentraSite Administration Tools

Notation

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

Term	Description
<SuiteInstallDir>	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/*.
<CentraSiteInstallDir>	This is the CentraSite installation directory. By default, this is the *CentraSite* folder under *<SuiteInstallDir>*.
<RuntimeDir>	This is the location of the Software AG Runtime. By default: *<SuiteInstallDir>/profiles/CTP*.

Term	Description
<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 *<SuiteInstallDir>\jvm\jvm<n>*, where *<n>* is the JDK version number.

1 Administering the License Key

- The Relationship Between the License Key and CentraSite Editions 2
- Changing the License Key 3
- Working with Time-Limited Licenses 3

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.

The following topics describes the purpose of the license key, and how to change the license key if required.

The Relationship Between the License Key and CentraSite 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.

For a description of the CentraSite editions and a list of the specific features in each, see the section *CentraSite Editions* in the document *Introducing CentraSite*.

How to Tell Which Edition of CentraSite You are Using

To determine which edition of CentraSite is running, open CentraSite Control and examine the banner at the top of the screen. 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 (e.g. to upgrade to a different edition of CentraSite).

To change a license key

1 Stop the CentraSite Registry Repository.

2 Identify the file system location where the file containing the current license key is stored. This is by default *<CentraSiteInstallDir>\lkey*.

3 Identify the file containing the current license key. This is by default *inm<nn>.xml*, where *<nn>* represents the product release number.

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.

4 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.

5 Start the CentraSite Registry Repository.

Working with Time-Limited Licenses

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

You can check the expiration date by examining the contents of the license key file, as follows:

To check the expiration date of your license

1 Open the license key file in a text editor.

2 Locate the element `ExpirationDate`.

3 If this element contains the value "Unlimited", there is no expiration date, i.e. 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.

2 Maintaining CentraSite's Internal Database

- Repository Monitoring 6
- Backing up the Database 8
- Restoring the Database from a Backup 9
- Deleting a Backup 11
- Moving a CentraSite Database to Another Location 12
- Locations 13
- Database Configuration Parameters 15
- Database Reorganization 16

The contents of the CentraSite Registry 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, using the techniques described in the following sections.

Repository Monitoring

While the CentraSite Registry 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.

- Database Activity
- Space Usage

Database Activity

To display the database activity information, use the following command:

```
inoadmin showactivity CentraSite
```

This displays the following items of 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.

Item	Description
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 "inoadmin showactivity CentraSite".
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 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/applications that are currently waiting for space allocation in the dynamic pool.
Total number of space waiters	The total number of users/applications that have waited for space allocation in the dynamic pool during the current session.

Space Usage

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 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. 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,

Database space	Description
	"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.

To display information about the database spaces, use the following command:

```
inoadmin listdbspaces CentraSite
```

For each database space, the following information is displayed:

- The type of database space, e.g. 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.

Backing up the Database

To protect against accidental loss of data, we recommend you to 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.

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. See the section **Locations** for information about predefined locations.

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, i.e., the state immediately after product installation, you can use this backup.

To back up the internal database

1 Decide whether you want to create the backup in the default backup location or in a different location. See the section **Locations** for information about defining a default backup location.

2 If you want to create the backup in the default backup location, using the following command:

```
inoadmin backup CentraSite
```

If you want to create the backup in a location other than the default backup location, using the following command:

```
inoadmin backup CentraSite <Location>
```

where <Location> is the path where the backup will be created, for example:

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

The location you specify must exist already, otherwise the backup will not run.

The Backup Key

When the backup completes, a status message is output 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

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.

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 choose 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, i.e., the state immediately after product installation, you can use this backup.

To restore the database from a backup

1 First make a backup of the current database, in case you decide to return to this database state at a later time.

2 If the CentraSite Registry Repository is running, stop it before continuing.

3 Decide which backup file you wish to use for the restore operation. You can show a list of all available backups and their backup keys (the unique identifiers for the backups) by using the command:

```
inoadmin listbackups CentraSite
```

4 Use a command of the following form to restore from the chosen backup:

```
inoadmin restore CentraSite <BackupKey>  <RecoverOption>
```

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 will be 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 will be restored to the state stored in the backup. No session log data will be 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` Note that 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, will also be retrieved from the session logs.

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

Disabled Backups

If you choose 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 will be set to disabled.

If you choose 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 will be 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, please 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, as described in the section **Moving a CentraSite Database to Another Location**, a database backup that originated on one machine can be configured so that it can be restored onto another machine.

Deleting a Backup

You can delete backups that are no longer required. Deleting a backup removes all of 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.

To delete a backup

1 Identify the backup that you wish to delete. You can list the available backups and their backup keys using the command:

```
inoadmin listbackups CentraSite
```

2 Enter the following command:

```
inoadmin deletebackup CentraSite <BackupKey>
```

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

Example:

```
inoadmin deletebackup CentraSite 001334732430
```

Note: 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. See the discussion on disabled backups in the section **Restoring the Database from a Backup** for more details.

Moving a CentraSite Database to Another Location

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:

1. if the architectures of the source and the target machine is with different byte orders (big-endian versus little-endian)
2. 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/repository contains environment-dependent data. This data has to be adjusted to the new environment when the data is moved to another machine.

The CentraSite kit contains a command line script that allows you to create a new database from an existing backup and to modify the data to the needs of the new environment. On Windows, the script is *MoveCentraSiteRR.cmd*, and on UNIX the script is *MoveCentraSiteRR.sh*.

The steps performed by the script 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

The script is called with 3 parameters as follows:

```
MoveCentraSiteRR.cmd  <BackupFilename>  <Username>  <Password>
```

The parameter *<BackupFilename>* is the name and path of the existing backup file. The script can only be run using the credentials of the predefined user "DefaultUser", so *<Username>* must be "DefaultUser" and *<Password>* must be the password of this user.

The script must be called from the *bin\cfg* location of the target CentraSite installation.

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

Example:

```
cd C:\SoftwareAG\CentraSite\bin\cfg
MoveCentraSiteRR.cmd C:\temp\CS820test.2BO DefaultUser PwdForDefUser
```

Locations

CentraSite uses certain default locations on disk to store information about the active CentraSite Registry 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.	*<CentraSiteInstallDir>\data*
backup	The Backup Location. This is the location where backup files are created by default.	*<CentraSiteInstallDir>\data*
log	The Log Location. This is the location where session log files are created by default. If the log archive location is defined (see next table entry), the log location holds only the most recent session log, and all other logs are stored in the log archive location.	*<CentraSiteInstallDir>\data*
archive	The Log Archive Location. This is the location where all session logs other than the current session log are stored.	(no default defined)

Location type	Purpose	Default path
	If no log archive location is defined, all session log files are held in the log location (see previous table entry), 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.	*<CentraSiteInstallDir>\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.

You can list the currently defined locations as follows:

To list the currently defined locations

- Use the following command:

```
inoadmin showlocations CentraSite
```

You can change the currently defined path of a location as follows:

To change the currently defined path of a location

- Use a command of the following form:

```
inoadmin setlocation CentraSite <LocationType> <path> [<path> ...] ↩
```

where *<LocationType>* is one of the location types shown in the table above. 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 "inoadmin 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 will be used as long as there is available disk space; if it is full, then D:\backuploc2 will be 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 will keep in parallel. When this number is exceeded, the oldest backup and the corresponding log spaces will be 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 will adjust the flush limit automatically.

For each property, the following information is available:

- **handle**
- **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**
- **configured**
- **current**
  The current value of the property.

- **type**
  The datatype of the property (string, numeric etc.).
- **unit**
  The unit of measurement for the property's value, e.g. megabytes or seconds.
- **state**

If you wish to examine or modify the database properties, you have the following possibilities:

Command	Description
`inoadmin listproperties CentraSite`	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, high water mark.
`inoadmin getproperty CentraSite <PropertyName>`	Show the value of the given property. If the property name contains one or more spaces, enclose the name in quotes.
`inoadmin setproperty CentraSite <PropertyName> <PropertyValue> [no]restart`	Change 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 will take effect at the next restart of the CentraSite Registry Repository. You can cause an automatic restart by specifying "restart", otherwise specify "norestart".
`inoadmin setproperties CentraSite <XMLInputFile> [no]restart`	Change 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 `inoadmin listproperties` command. The changed values will take effect at the next restart of the CentraSite Registry Repository. You can cause an automatic restart by specifying "restart", otherwise specify "norestart".

Database Reorganization

Within the CentraSite's internal database a considerable amount of data may be stored temporarily, e.g. 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 will remain. 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 in a way that unused space can be de-allocated.

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 will be deleted when the reorganize function completes successfully.

To execute database reorganization, use the following command:

```
inoadmin reorganize CentraSite
```

The `reorganize` command reduces the diskspace by returning blocks no longer used to the file system.

To check the available free disk space, execute the following command:

```
inoadmin reorganize CentraSite evaluate
```

The `evaluate` command returns the current amount of free disk space in the form `INODSI1183: 161.84 MB empty data space found`.

Remember that for the `reorganize` command, the database must be stopped beforehand and re-started afterwards as execution of this command against an active database will be denied. However, the database need not be stopped when using the `evaluate` option.

3 Configuring the Authentication Settings

- Listing details of a particular configuration 20
- Setting the default configuration 23
- Adding a configuration 23
- Modifying a configuration 26
- Removing a configuration 27
- Validating a configuration 28

The authentication in the CentraSite Registry 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 will be 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.

You can view and modify the authentication settings using the command line tool *CentraSiteCommand.cmd* (Windows) or *CentraSiteCommand.sh* (UNIX). The tool is located in `<CentraSiteInstallDir>`/utilities.

The parameters of the command are case-sensitive, so for example the parameter "-url" must be specified as shown and not as "-URL".

Listing details of a particular configuration

To list details of a particular configuration, use a command of the following form:

```
CentraSiteCommand get Authentication [-url <CENTRASITE-URL>] -user <USER-ID>
-password <PASSWORD> -domain <DOMAIN>
```

The following table describes the complete set of input parameters that you can use with the `get Authentication` utility:

Parameter	Description
`-url`	The fully qualified URL (http://localhost:53305/CentraSite/CentraSite or http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
`-user`	The user ID of a user who has the "CentraSite Administrator" role.
`-password`	The password of the user identified by the parameter "-user".
`-domain`	The domain name of the user repository associated with the configuration.

For example:

```
CentraSiteCommand get Authentication -url ↩
"http://localhost:53307/CentraSite/CentraSite"
-user "Administrator" -password "manage" -domain "LDAPDomain"
```

The details are returned as an XML file. The XML file has a root element `ino:domain` that has the following attributes:

Element name	Description
ino:acceptusers	Meaning: This specifies whether to allow access of any user that is correctly authenticated by the authentication service or whether to only allow access by users that are explicitly defined in CentraSite. Possible values: "all" - Allow access of any user that is correctly authenticated by the authentication service; "defined" (default value) - allow access only to users defined in CentraSite
ino:casesensitiveuserids	Meaning: This determines whether or not user names in this domain are case-sensitive. Possible values: true - user IDs in this domain are case-sensitive; false - user IDs are not case-sensitive
ino:default	Meaning: This determines whether or not the configuration is the default configuration. Possible values: true - This is the default configuration; false - this is not the default configuration.
ino:domainid	Meaning: The domain name of the user repository associated with the configuration.
ino:domaintype	Meaning: The type of user repository associated with the configuration. Possible values: Typical values are: "INTERNAL" (the default domain), or a Windows domain name or an LDAP domain name.
ino:expire	Meaning: The amount of time (in seconds) that the user is cached in the server after successful authentication. Changes made to the user, e.g. deletion or password changes, do not take effect until this time has elapsed. The default is 120 seconds. This setting is provided for performance reasons. A value of 120 seconds is reasonable. If the connection to the LDAP server is slow, you can increase this figure.
ino:usegroups	Meaning: This specifies whether to use the external group information from domains; for example, the groups in an Active Directory Server or in an LDAP server. Possible values: true - use external group information; false (default value) - do not use external group information.

Example

Here is an example of an authentication configuration returned as an XML file:

```
<ino:domain xmlns:ino="http://namespaces.softwareag.com/tamino/response2"
ino:acceptusers="all" ino:casesensitiveuserids="false" ino:default="false"
ino:domainid="LDAP" ino:domaintype="ldap" ino:expire="120" ino:usegroups="true">
  <ino:param ino:content="ldap://ldapserver12" ino:name="host"/>
  <ino:param ino:content="10389" ino:name="port"/>
  <ino:param ino:content="ApacheDS" ino:name="ldap_server_type"/>
  <ino:param ino:content="ou=people,ou=RegionNorth,o=WidgetCo" ↩
ino:name="ldap_person_dn"/>
  <ino:param ino:content="inetOrgPerson" ino:name="ldap_person_object"/>
  <ino:param ino:content="cn" ino:name="ldap_user_field"/>
  <ino:param ino:content="ou=groups,ou=RegionNorth,o=WidgetCo" ↩
ino:name="ldap_group_dn"/>
  <ino:param ino:content="groupOfUniqueNames" ino:name="ldap_group_object"/>
  <ino:param ino:content="uniqueMember" ino:name="ldap_group_person_attribute"/>
  <ino:param ino:content="rd" ino:name="ldap_resolve_groups"/>
  <ino:param ino:content="TRUE" ino:name="useLdapTechUser" />
  <ino:param ino:content="c:\softwareag\centrasite\bin\cred.txt" ↩
ino:name="techLdapUserCredFile" />
  <ino:param ino:content="c:\softwareag\centrasite\bin\key.txt" ↩
ino:name="techLdapUserKeyFile" />
  <ino:configuration>
    <ino:group>
       <ino:properties>
           <ino:mapping ino:external="description" ino:local="description"/>
       </ino:properties>
    </ino:group>
    <ino:user>
      <ino:properties>
         <ino:mapping ino:local="organization" ino:external="org"/>
         <ino:mapping ino:local="emailAddresses:emailAddress:address" ↩
ino:external="mail"/>
         <ino:mapping ino:local="telephoneNumbers:telephoneNumber:number" ↩
ino:external="telephoneNumber"/>
         <ino:mapping ino:local="telephoneNumbers:telephoneNumber:countryCode" ↩
ino:external="telephoneCode"/>
         <ino:mapping ino:local="telephoneNumbers:telephoneNumber:extension" ↩
ino:external="telephoneExt"/>
         <ino:mapping ino:local="telephoneNumbers:telephoneNumber:areaCode" ↩
ino:external="telephoneAreaCode"/>
         <ino:mapping ino:local="personName:firstName" ino:external="cn"/>
         <ino:mapping ino:local="description" ino:external="description"/>
         <ino:mapping ino:local="postalAddresses:postalAddress:postalCode:" ↩
ino:external="postalcode"/>
         <ino:mapping ino:local="postalAddresses:postalAddress:city:" ↩
ino:external="postalcity"/>
         <ino:mapping ino:local="postalAddresses:postalAddress:stateOrProvince" ↩
ino:external="stateorprovince"/>
         <ino:mapping ino:local="postalAddresses:postalAddress:country" ↩
ino:external="countrycode"/>
         <ino:mapping ino:local="URL" ino:external="E-mail"/>
      </ino:properties>
    </ino:user>
```

```
  </ino:configuration>
</ino:domain>
```

For details of the meaning of fields that are required for the configuration, see the document *Authentication Topics and LDAP*.

Setting the default configuration

To set the default configuration, use a command of the following form:

```
CentraSiteCommand set DefaultDomain [-url <CENTRASITE-URL>] -user
<USER-ID> -password <PASSWORD> -domain <DOMAIN>
```

The following table describes the complete set of input parameters that you can use with the `set DefaultDomain` utility:

Parameter	Description
`-url`	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
`-user`	The user ID of a user who has the "CentraSite Administrator" role.
`-password`	The password of the user identified by the parameter "-user".
`-domain`	The domain name of the user repository associated with the configuration.

For example:

```
CentraSiteCommand set DefaultDomain -url ↩
"http://localhost:53307/CentraSite/CentraSite"
-user "Administrator" -password "manage" -domain "LDAPdomain"
```

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

Adding a configuration

You can add a configuration by using one of the following methods:

- specifying a configuration file containing a complete configuration as input to CentraSiteCommand
- using CentraSiteCommand's interactive wizard
- specifying a LDAP domain name

Tip: For the meaning of fields that are required for the configuration, see the document *Authentication Topics and LDAP*.

Adding a Configuration using a Configuration File

To add a configuration using a configuration file, use a command of the following form:

```
CentraSiteCommand set Authentication [-url <CENTRASITE-URL>] -user <USER-ID>
-password <PASSWORD> -file <CONFIG-FILE>
```

The following table describes the complete set of input parameters that you can use with the `set Authentication` utility:

Parameter	Description
`-url`	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
`-user`	The user ID of a user who has the "CentraSite Administrator" role.
`-password`	The password of the user identified by the parameter "-user".
`-file`	The URI (file: or http:) of the configuration file.

For example:

```
CentraSiteCommand set Authentication -url ↩
"http://localhost:53307/CentraSite/CentraSite"
-user "Administrator" -password "manage" -file "config.xml"
```

To create the XML configuration file, you can use the `get Authentication` utility described above to retrieve an existing configuration as an XML file, then modify the entries as required.

Adding a Configuration using CentraSiteCommand's Interactive Wizard

To add a configuration using CentraSiteCommand's interactive wizard

1
```
CentraSiteCommand set Authentication [-url <CENTRASITE-URL>] -user <USER-ID>
-password <PASSWORD>
```

The following table describes the complete set of input parameters that you can use with the `set Authentication` utility:

Parameter	Description
`-url`	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
`-user`	The user ID of a user who has the "CentraSite Administrator" role.
`-password`	The password of the user identified by the parameter "-user".
`-file`	The URI (file: or http:) of the configuration file.

For example:

```
CentraSiteCommand set Authentication -url ↩
"http://localhost:53307/CentraSite/CentraSite"
-user "Administrator" -password "manage"
```

2 Follow the steps in the wizard to define the LDAP configuration. The wizard takes you through a set of dialogs in order to define the following information:

- Basic LDAP Host configuration
- User configuration
- User information mapping
- Group configuration
- Group information mapping
- Group resolution configuration
- The domain ID of the configuration

Tip: For the meaning of fields that are required for the configuration, see the document *Authentication Topics and LDAP*.

Adding a Configuration using an LDAP Domain Name

To add a configuration using a LDAP domain name, use a command of the following form:

```
CentraSiteCommand set Authentication [-url <CENTRASITE-URL>] -user <USER-ID>
-password <PASSWORD> [-domain <DOMAIN>]
```

The following table describes the complete set of input parameters that you can use with the `set Authentication` utility:

Parameter	Description
`-url`	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
`-user`	The user ID of a user who has the "CentraSite Administrator" role.
`-password`	The password of the user identified by the parameter "-user".
`-domain`	The domain name of the user repository associated with the configuration.

For example:

```
CentraSiteCommand set Authentication -url ↩
"http://localhost:53307/CentraSite/CentraSite"
-user "Administrator" -password "manage" -domain LDAPdomain
```

Modifying a configuration

To modify a configuration, use a command of the following form:

```
CentraSiteCommand set Authentication [-url <CENTRASITE-URL>] -user <USER-ID>
-password <PASSWORD> [-domain <DOMAIN>]
```

The following table describes the complete set of input parameters that you can use with the `set Authentication` utility:

Parameter	Description
`-url`	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
`-user`	The user ID of a user who has the "CentraSite Administrator" role.
`-password`	The password of the user identified by the parameter "-user".
`-domain`	The domain name of the user repository associated with the configuration.

For example:

```
CentraSiteCommand set Authentication -url ↩
"http://localhost:53307/CentraSite/CentraSite"
-user "Administrator" -password "manage" -domain LDAPdomain
```

This command invokes a command line wizard that runs through the same steps as the wizard for adding a configuration. The wizard displays the stored values for the configuration's fields and allows you to enter new values if required.

Removing a configuration

To remove a configuration, use a command of the following form:

```
CentraSiteCommand remove Authentication [-url <CENTRASITE-URL>] -user
<USER-ID> -password <PASSWORD> -domain <DOMAIN>
```

The following table describes the complete set of input parameters that you can use with the `remove Authentication` utility:

Parameter	Description
`-url`	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
`-user`	The user ID of a user who has the "CentraSite Administrator" role.
`-password`	The password of the user identified by the parameter "-user".
`-domain`	The domain name of the user repository associated with the configuration.

For example:

```
CentraSiteCommand remove Authentication -url ↩
"http://localhost:53307/CentraSite/CentraSite"
-user "Administrator" -password "manage" -domain "LDAPdomain"
```

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.

Validating a configuration

You can use a validation command to check whether the configuration is set up correctly and can be used to log in. A domain user name and password must be specified additionally to validate the domain.

To validate a configuration, use a command of the following form:

```
CentraSiteCommand validate Authentication [-url <CENTRASITE-URL>] -user
<USER-ID> -password <PASSWORD> -domain <DOMAIN>
```

The following table describes the complete set of input parameters that you can use with the `validate Authentication` utility:

Parameter	Description
`-url`	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
`-user`	The user ID of a user who has the "CentraSite Administrator" role.
`-password`	The password of the user identified by the parameter "-user".
`-domain`	The domain name of the user repository associated with the configuration.

For example:

```
CentraSiteCommand validate Authentication -url ↩
"http://localhost:53307/CentraSite/CentraSite"
-user "Administrator" -password "manage" -domain "domain"
```

4 Configuring Port Numbers

- Changing the Port Numbers of the CentraSite Registry/Repository 30
- Changing the Software AG Runtime Port Numbers 31

This chapter gives information about HTTP/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.

The document contains the following sections:

Changing the Port Numbers of the CentraSite Registry/Repository

The CentraSite Registry Repository uses several port numbers that have the following default values:

Port name	Description	Default port number
XML XTS port	The application port.	53301
ADMIN port	The administration port. The port used for access to the CentraSite Registry Repository server for all administration actions.	53303
HTTP port	The HTTP Server port. The TCP/IP port used for HTTP access to the CentraSite Registry Repository.	53313

If you wish to change these port numbers, the following locations must be updated:

- The registry of the machine on which the CentraSite Registry 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.

To change a port number on the CentraSite Registry Repository host

1 Run the command line script *centrasite_setenv*. This script is located in `<CentraSiteInstallDir>`*/bin*.

This ensures that environment variables and lookup paths are set correctly for the following steps.

2 Check the current value of the port number by using a command of the following form at the operating system command prompt:

```
inoadmin getproperty CentraSite "<ParameterName>"
```

The <ParameterName> can be any of the port names shown in the above table. Example:

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

The *inoadmin* program is available under the location <CentraSiteInstallDir>/*bin*.

3 Assign a new port number by using the following command at the operating system command prompt:

```
inoadmin setproperty CentraSite "<ParameterName>" "<NewPortNumber>" norestart
```

where <NewPortNumber> is the new port number that you wish to use.

4 Stop the CentraSite Registry Repository and start it again.

The procedure to change a configured CRR port in a CAST web application is as follows:

To change a CRR port number in a CAST web application

1 Open the configuration file of the web application.

See the section **Changing the Certificate Configuration for the CAST Components** for information about the names and locations of the configuration files for the web applications.

Note that not all of the web applications store information about the CRR port numbers.

2 Search for a string containing a URL of the form *https://<machine>:<port>/CentraSite/CentraSite*, where <machine> is the machine hosting the CRR and <port> is the currently used CRR port.

3 Replace the port number by the new port number.

4 If there are any other such URLs in the file, modify them also as required.

Ensure that you change the configuration files of all of the CAST web applications that contain CRR port numbers. When you have done this, restart the Software AG Runtime.

Changing the 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 53305 is used to provide compatibility with the previous product releases as it was used by the CRR.

The port numbers are configured in property files that are located in <SuiteInstallDir>/*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 in this location 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.

In general, Software AG Runtime has to be restarted for the changes to take effect.

5 Configuring Secure Communication between CentraSite Components

- Secure Communication between the CRR and the CAST 34
- Secure Communication between Software AG Runtime and External Clients 39

This chapter 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.

The document contains the following sections:

Secure Communication between the CRR and the CAST

The communication between the CRR and the CAST components takes place via 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 deactivate the SSL communication between the CRR and the CAST components, as described in the subsequent section **Allowing HTTP Communication between CAST and CRR**. However, Software AG strongly recommends you NOT to do this, because it opens a potential security risk.

You can configure aspects of the SSL setup, as described in the following sections.

- Changing the Certificate Configuration for the Registry Repository
- Changing the Certificate Configuration for the CAST Components
- The CAST Stores

- Allowing HTTP Communication between CAST and CRR

Changing the Certificate Configuration for the Registry Repository

The CRR provides the following configurable properties:

Property name	Purpose
SSL certificate file	Name of the file that contains the server certificate. The default is *<CentraSiteInstallDir>/files/certs/crrcert.crt*.
SSL key file	Name of the file that contains the private server key. The default is *<CentraSiteInstallDir>/files/certs/crr.key*.
SSL password	Password for accessing the SSL configuration files. The default is "cscert".
SSL CA file	Name of the file that contains the certificate authority (CA) truststore. The default is *<CentraSiteInstallDir>/files/certs/cstrust.pem*. This file would normally contain the client certificate but actually contains the CA certificate and key.
SSL verify client	Perform client authentication during handshake. Possibly values are "yes" and "no". The default is "yes".
SSL verify depth	Depth of certificate chain used for client authentication. The default is 1.

The key and certificate files need to be in an OpenSSL readable format. The CA file needs to be in PEM format.

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

The server parameters can be changed via the command line tool inoadmin.

The general syntax is

```
inoadmin setproperty CentraSite "<PropertyName>" "<PropertyValue>" norestart
```

For example:

```
inoadmin setproperty CentraSite "SSL certificate file" ↩
"C:/SoftwareAG/CentraSite/files/certs/custom_cacert.pem" norestart
```

Restart the CRR after changing the parameter settings.

Changing the Certificate Configuration for the CAST Components

The CAST web applications read the SSL configuration from their deployment descriptor, which is located at *<CentraSiteInstallDir>/cast/cswebapps/<WebApplicationName>/WEB-INF/web.xml*. For some of these web applications, you can change the SSL settings in the *web.xml* files. The web applications for which this applies are:

- CentraSite
- Centrasite_authenticated (this web application is disabled by default for security reasons)
- SOALinkSNMPEventsListener
- UddiRegistry
- BusinessUI

For the CentraSiteControl application, the SSL configuration is stored in *<RuntimeWebAppsDir>/PluggableUI/CentraSiteControl/plugin.xml*.

For the BusinessUI application, the SSL configuration is stored in *<CentraSiteInstallDir>/cast/cswebapps/BusinessUI/system/conf/centrasite.xml*.

The *web.xml* configuration files contain entries like the following. Modify the <param-value> values as desired, then restart the Software AG Runtime.

```
<init-param>
  <param-name>com.softwareag.centrasite.security.trustStore</param-name>
  <param-value>C:/SoftwareAG/CentraSite/cast/files/certs/casttrust.p12</param-value>
</init-param>
<init-param>
  <param-name>com.softwareag.centrasite.security.trustStorePassword</param-name>
  <param-value>cscert</param-value>
</init-param>
<init-param>
  <param-name>com.softwareag.centrasite.security.trustStoreType</param-name>
  <param-value>PKCS12</param-value>
</init-param>
<init-param>
  <param-name>com.softwareag.centrasite.security.keyStore</param-name>
  <param-value>C:/SoftwareAG/CentraSite/cast/files/certs/castcert.p12</param-value>
</init-param>
<init-param>
  <param-name>com.softwareag.centrasite.security.keyStorePassword</param-name>
  <param-value>cscert</param-value>
</init-param>
<init-param>
  <param-name>com.softwareag.centrasite.security.keyStoreType</param-name>
  <param-value>PKCS12</param-value>
</init-param>
```

The meaning of the properties corresponds to the system properties of the Java 2 platform package "javax.net.ssl":

- javax.net.ssl.trustStore
- javax.net.ssl.trustStorePassword
- javax.net.ssl.trustStoreType
- javax.net.ssl.keyStore
- javax.net.ssl.keyStorePassword
- javax.net.ssl.keyStoreType

For the CentraSiteControl application, the file *plugin.xml* contains entries of the form `<extension ... id="..." value="...">`. The `id` and `value` entries correspond to the `param-name` and `param-value` entries of the `web.xml` files.

For the BusinessUI application, the SSL settings are defined in the element `<SSL>` in the *centrasite.xml* file, using the same property naming conventions as in the `web.xml` files.

The CAST Stores

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

- The keystore certificate. This is located at *<CentraSiteInstallDir>/cast/files/certs/castcert.p12*. It contains the client certificate and private client key.
- The truststore certificate. This is located at *<CentraSiteInstallDir>/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.

Allowing HTTP Communication between CAST and CRR

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

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, please consider carefully that 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.

If you wish to use a mixed HTTP/HTTPS communication, proceed as follows:

To allow mixed HTTP/HTTPS communication between CAST and CRR

1 Use inoadmin to change the communication method setting as follows:

```
inoadmin setproperty CentraSite "communication method" "HTTP and HTTPS" restart
```

2 Make the following change in *<CentraSiteInstallDir>/cast/cswebapps/CentraSite/WEB-INF/web.xml*:

Change the value of `com.softwareag.centrasite.sslusage` from "yes" to "no".

3 Make the following change in *<CentraSiteInstallDir>/cast/cswebapps/CentraSite_authenticated/WEB-INF/web.xml.disabled*:

Change the value of `com.softwareag.centrasite.sslusage` from "yes" to "no".

4 Make the following changes in *<CentraSiteInstallDir>/cast/cswebapps/SOALinkSNMPEventsListener/WEB-INF/web.xml*:

Change the value of `com.softwareag.centrasite.sslusage` from "yes" to "no".

Change the value of `com.softwareag.centrasite.soalink.events.dbUrl` to use "http" instead of "https".

5 Make the following changes in *<CentraSiteInstallDir>/cast/cswebapps/UddiRegistry/WEB-INF/web.xml*:

Change the value of `com.softwareag.centrasite.sslusage` from "yes" to "no".

Change the value of `com.centrasite.uddi.store.db` to use "http" instead of "https".

6 Make the following changes in *<CentraSiteInstallDir>/cast/cswebapps/BusinessUI/system/conf/centrasite.xml*:

Change the `url` attribute of the `CentraSite` element to use "http" instead of "https".

Change the value of the `sslusage` attribute of the `SSL` element from "yes" to "no".

7 Make the following changes in *<RuntimeDir>/workspace/webapps/PluggableUI/CentraSiteControl/plugin.xml*:

Change the value of `com.softwareag.centrasite.sslusage` from "yes" to "no".

Change the value of `crrUrl` to use "http" instead of "https".

Secure Communication between Software AG Runtime and External Clients

- Overview
- The Software AG Runtime properties file for SSL communication
- The SSL keystore
- The SSL truststore
- Note on SSL port number

Overview

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

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

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

Please consult the Tomcat manuals for details on how to configure the SSL-based authentication – here we only provide the basics. General instructions on how to protect Tomcat can be found under links such as the following (version-specific for Tomcat 7.0):

- **http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html**
- **http://tomcat.apache.org/tomcat-7.0-doc/config/http.html#SSL_Support**

The Software AG Runtime properties file for SSL communication

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

Refer also to the general cross-product instructions for setting Software AG Runtime properties for SSL communication at **http://documentation.softwareag.com/webmethods/wmsuites/wmsuite9-6/Cross_Product/9-6_Working_with_Runtime.pdf**, which describes how to configure 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*.

The SSL keystore

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

Please 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 that 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"). Please set the keystore password accordingly (parameter `keystorePass` in the Software AG Runtime properties file).

The 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

For a full description of these properties, refer to the Tomcat SSL documentation at **http://tomcat.apache.org/tomcat-7.0-doc/config/http.html**.

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.

6 Configuring the Registry Cache Settings

- Prerequisites 42
- Displaying The Cache Configuration 42
- Modifying The Cache Configuration 43

For performance reasons, CentraSite uses an internal cache to store registry objects accessed via JAXR. In some cases, you might want to modify the cache configuration settings, in order to determine the optimal setup for your system. This section describes a command line tool that you can use for such purposes.

This command line tool allows you to display and to modify the JAXR-based configuration settings. The JAXR-based configuration settings apply to each connection.

The tool consists of an executable jar file *CentraSiteCacheConfiguration.jar* that is located in the *bin* folder of the CentraSite installation.

Prerequisites

To be able to use the command line tool, please note the following points:

- The user specified in the command line must have the "CentraSite Administrator" role.
- The CentraSite Registry Repository must be online.
- The tool requires a Java 6 runtime.

Displaying The Cache Configuration

In order to display the cache configuration, run the tool with the DISPLAY keyword.

To display the cache settings for JAXR

- ```
  java -jar CentraSiteCacheConfiguration.jar <CentraSite DB URL>
            <administrator user id> <password> DISPLAY
     ↩
  ```

  for example:

  ```
  java -jar CentraSiteCacheConfiguration.jar ↩
  "http://localhost:53307/CentraSite/CentraSite"
            DOMAIN\admin pAsSwOrD DISPLAY
  ```
  ```

Modifying The Cache Configuration

The SET keyword is followed by pairs of option and value. After the modification operation is completed, the tool will display the modified cache configuration. The following options may be specified:

- maxElementsOnHeap
- maxElementsOffHeap
- memoryStoreEvictionPolicy
- statistics

If one of the options is not specified in the SET operation, its existing value will be copied.

The SET keyword is used as follows:

To modify the cache settings for JAXR

- ```
  java -jar CentraSiteCacheConfiguration.jar <CentraSite DB URL>
            <administrator user id> <password> SET <option> <value> ↩
  [<option> <value>  ...]
  ```

  for example:

  ```
  java -jar CentraSiteCacheConfiguration.jar ↩
  "http://localhost:53307/CentraSite/CentraSite"
            DOMAIN\admin pAsSwOrD SET maxElementsOnHeap 10000
  ```

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 will be evicted according to the algorithm specified by the `memoryStoreEvictionPolicy` option.
  ```

Option: maxElementsOffHeap

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

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.

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

7 Configuring the Email Server

- Configuring the Email Server Settings 46
- Getting the Email Server Settings 48

Certain facilities within CentraSite communicate information to users using email (the Send Email Notification policy action, for example). These facilities will 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 the Email Server Settings

Use the following procedure to specify the email server settings for CentraSite. To perform this procedure, 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.

You can configure the email server settings by executing the following commands in the command line interface *CentraSiteCommand.cmd* (Windows) or *CentraSiteCommand.sh* (UNIX) of Command Central. The tool is located in *<CentraSiteInstallDir>*/utilities.

If you start this command line 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".

To configure the email server settings

1 Create an XML configuration file that contains the following predefined properties. This file should be in Java XML properties format. For 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>
```

Descriptions of these properties are as follows:

In this field...	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 will wait 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".
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 configured so that the client will trust the server's certificate. Defaults to false.

2 Execute the following command in this format:

```
CentraSiteCommand set Email [-url <CENTRASITE-URL>] -user <USER-ID> -password
<PASSWORD> -file <CONFIG-FILE>
```

The following table describes the complete set of input parameters that you can use with the `set Email` utility:

Parameter	Description
`-url`	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
`-user`	The user ID of a user who has the "CentraSite Administrator" role.
`-password`	The password of the user identified by the parameter "-user".
`-file`	The URI (file: or http:) of the configuration file.

For example:

```
CentraSiteCommand set Email [-url "http://localhost:53307/CentraSite/CentraSite"] ↩
-user "Administrator" -password
"manage" -file "config.xml"
```

Getting the Email Server Settings

You can get (retrieve) the email server settings by executing a command in the command line interface of Command Central.

To get the settings, execute the following command, which will print the configuration:

```
CentraSiteCommand.sh get Email [-url <CENTRASITE-URL>] -user <USER-ID> -password
<PASSWORD> -file <CONFIG-FILE>
```

The following table describes the complete set of input parameters that you can use with the `get Email` utility:

Parameter	Description
`-url`	The fully qualified URL (http://localhost:53307/CentraSite/CentraSite) for the CentraSite registry/repository.
`-user`	The user ID of a user who has the "CentraSite Administrator" role.
`-password`	The password of the user identified by the parameter "-user".
`-file`	The URI (file: or http:) of the configuration file.

For example:

```
CentraSiteCommand.sh get Email [-url "http://localhost:53307/CentraSite/CentraSite"] ↩
-user "Administrator" -password
"manage" -file "config.xml"
```

8 Overview of the CentraSite Administration Tools

- Overview of the Command Line Tool "inoadmin" 52
- Overview of Command Central 53
- Return Codes from Command Execution 53

Overview of the Command Line Tool "inoadmin"

The command line tool "inoadmin" offers functionality for performing low-level administrative operations on the CentraSite Registry Repository. The functionality provided by inoadmin 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.

The practical use of the tool is described in the scenarios documented in the preceding sections.

The tool is located in the directory <*CentraSiteInstallDir*>*bin*. If you call inoadmin without parameters, you receive a summary of the command syntax and the available functions.

You can use environment variables to configure the inoadmin behavior:

Environment variable	Purpose
INOADMIN_OUTPUT_TXT	Determines whether the inoadmin output is displayed as an XML document or as a formatted table. If the value is null (i.e. 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 (i.e. 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.
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.

Overview of Command Central

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

- Infrastructure engineers can see at a glance which products and fixes are installed where, and can easily compare installations to find discrepancies.
- System administrators can configure environments using a single web UI, command-line tool or API, so maintenance can be performed with a minimum of 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

For more information, see the Command Central documentation.

Return Codes from 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.

Return Code	Description
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.
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 function called is not implemented.