CA IT Client Manager

Software Delivery CLI Reference Guide

Release 12.8

CA Technologies Product References

This documentation set references to the following CA products:

- CA Advantage® Data Transport® (CA Data Transport)
- CA Asset Intelligence
- CA Asset Portfolio Management (CA APM)
- CA Common Services™
- CA Desktop Migration Manager (CA DMM)
- CA Embedded Entitlements Manager (CA EEM)
- CA Network and Systems Management (CA NSM)
- CA Patch Manager
- CA Process Automation
- CA Business Intelligence
- CA Service Desk Manager
- CA WorldView™
- CleverPath™ Reporter

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Overview

The *Software Delivery CLI Reference Guide* provides detailed reference information for three separate software delivery command line interfaces:

- sd_acmd
- sd_msiexe
- sd_sscmd

Chapter 2: sd_acmd---Agent Administrative Commands

The sd_acmd command reports on manually installed software.

Software Delivery agent software must be installed and configured before you can use sd_acmd.

You can execute sd_acmd from your job script to:

- Signal reboots or logoffs
- Populate install or uninstall records
- Populate or remove scalability server staging library delivery records
- Populate or remove software detection records
- Secure and execute Software Delivery job container order files.

Syntax

`sd_acmd` *`command command_arguments [arguments]`*

Note: For offline reinstallation, you can use the ReinstallTarget, Encrypt, and JobCheck commands. These commands are listed and explained in the following sections.

This section contains the following topics:

AddActivateRecord--Software Package Was Activated

This command adds a record notifying Software Delivery that a software package has been activated.

Syntax

```
AddActivateRecord item
                  version
                  procedure
                  installprocedure
                  date
                  time
                  orderedby
                  comment
                   [target=<target>]
```

item

Name of the software package.

version

Version of the software package.

procedure

Blank string (reserved for future use).

installprocedure

Name of the installation procedure used. Use a blank string to indicate any install procedure, and an asterisk (*) to indicate every installation made, using any install procedure.

date

Specifies the date of the installation. You can also use the keyword **current**. Format of date: YYYY-MM-DD

time

Specifies the time of the installation. You can also use the keyword **current**. Format of time: HH:MM

orderedby

Specifies the User ID that ordered the installation.

comment

Your own comments.

[target=<target>]

Specifies whether the command is applicable for the computer unit (default), or any valid or existing user profile units on the same computer. You can specify any target except the following targets:

- On the computer that you are running the command.
- From the user profile on the same computer you are running the command on.

Example

In the following example Software Delivery is notified that an activation record for software item Product X version 1.1, using procedure Activate1, with reference to install procedure Install, should be added. The current date and time are used for the installation. Ordered by will list Automatic by Job and the comment will be Using Activate1 for Product X 1.1:

```
sd_acmd addactivaterecord "Product X" "1.1" "Activate1" "Install" current current
"Automatic by Job" "Using Activate1 for Product X 1.1"
```

Note: To omit an argument, include the empty argument in your syntax by typing "".

If there is no installation that the activation record can be associated with, the result of the command is not saved.

AddConfigureRecord--Software Package Was Configured

This command adds a record notifying Software Delivery that a software package has been configured. If there is no installation that the configuration record can be associated with, the result of the command is not saved.

Syntax

```
AddConfigureRecord item
                   version
                   procedure
                   installprocedure
                   date  | current
                   time  | current
                   orderedby
                   comment
                   [target="targetname"]
```

item

Name of the software package.

version

Version of the software package.

procedure

Blank string (reserved for future use).

installprocedure

Name of the installation procedure used. Use a blank string to indicate any install procedure, and an asterisk (*) to indicate every installation made, using any install procedure.

date

Specifies the date of the installation. You can also use the keyword **current**. Format of date: YYYY-MM-DD

time

Specifies the time of the installation. You can also use the keyword **current**. Format of time: HH:MM

orderedby

This parameter is optional and provides an account name, for example, DomainX\UserY.
Default: Current user running the SecureContainer command.

comment

Your own comments.

[target="targetname"]

Specifies whether the command is applicable for the computer unit (default), or any valid or existing user profile units on the same computer. You can specify any target except the following targets:

- On the computer that you are running the command.
- From the user profile on the same computer you are running the command on.

Example

In the following example Software Delivery is notified that a configuration record for software item Product X Version 1.1, using procedure Configure1, with reference to install procedure Install, should be added. The current date and time are used for the installation. Ordered by will list Automatic by Job and the comment will be Using Configure1 for Product X 1.1:

```
sd_acmd addconfigurerecord "Product X" "1.1" "Configure1" "Install" current current
"Automatic by Job" "Using Configure1 for Product X 1.1"
```

Note: To omit an argument, include the empty argument in your syntax by typing "".

AddDetectedRecord--Notify Software Delivery that a Software Package Was Detected

This command adds a record notifying Software Delivery that a software package has been detected. The difference between an installed and a detected package is that the installed package has source files available in the Library while the detected one has no such files.

Syntax

```
AddDetectedRecord item
                  version
                  supplier
                  comment
                 [uninstall template]
                 [uninstall file]
                 [uninstall params]
                 [target="targetname"]
```

item

Name of the software package.

version

Version of the software package.

supplier

Name of the supplier.

comment

Your own comments.

uninstall_template

Template type, for example *msidetup* for MSI packages.

uninstall_file

The name of the uninstall file, for example *x.msi* for MSI packages.

uninstall_params

Uninstall parameters, for example */u $joid /x productGUID /lemo $rf /qn* for MSI packages.
The macro *$joid* will expand to the job object identifier.
The macro *$rf* will create a result file to enable view of the output.
The *x* parameter states that an uninstall should be made.
The *qn* parameter states that the execution should occur silently, that is, without a user interface.

[target="targetname"]

Specifies whether the command is applicable for the computer unit (default), or any valid or existing user profile units on the same computer. You can specify any target except the following targets:

- On the computer that you are running the command.
- From the user profile on the same computer you are running the command on.

Example

In the following example:

```
sd_acmd adddetectedrecord "Product V" "3.1" "Bestseller Inc." "Detection of Product
V v3.1" "" "" ""
```

Software Delivery is told that a record for detected software item Product V Version 3.1 should be added. Supplier will list Bestseller Inc. and the comment will be "Detection of Product V v3.1."

Notes: If you want to omit arguments, say the reserved arguments, you still have to include the empty arguments in your list by typing "".

For SXP, PIF, PKG and RPM packages, if an AddDetectedRecord is issued, and the Item is registered in the software library, the detection record is automatically converted to an installation record with procedure Install Package.

AddInstallRecord--Notify Software Delivery that a Software Package Was Installed

This command adds a record notifying Software Delivery that a software package has been installed. If an AddInstallRecord is issued, and the Item or Item Procedure is missing in the software library, the installation record is automatically converted to a detection record with procedure Detected.

Syntax

```
AddInstallRecord item
                 version
                 procedure
                 date | current
                 time | current
                 orderedby
                 comment
                 [target="targetname"]
```

item

Name of the software package.

version

Version of the software package.

procedure

Blank string (reserved for future use).

date

Specifies the date of the installation. You can also use the keyword **current**.
Format of date: YYYY-MM-DD

time

Specifies the time of the installation. You can also use the keyword **current**.
Format of time: HH:MM

orderedby

This parameter is optional and provides an account name, for example, DomainX\UserY.
Default: Current user running the SecureContainer command.

comment

Your own comments.

[target="targetname"]

Specifies whether the command is applicable for the computer unit (default), or any valid or existing user profile units on the same computer. You can specify any target except the following targets:

- On the computer that you are running the command.
- From the user profile on the same computer you are running the command on.

Example

In the following example Software Delivery is notified that an installation record for software item Product X Version 1.1, using procedure Install, should be added. The current date and time are used for the installation. Ordered by will list Automatic by Job and the comment will be required by Product Y v2.0 install.

```
sd_acmd addinstallrecord "Product X" "1.1" "Install" current current "Automatic by Job" "Required by Product Y v2.0 install"
```

Note To omit an argument, include the empty argument in your list by typing "".

AddLibDeliveryRecord--Notify Software Delivery that a Software Package Was Added to a scalability server staging library

This command adds a record notifying Software Delivery that a software package has been added to a scalability server staging library.

Syntax

```
AddLibDeliveryRecord item
                     version
                     procedure
                     date  | current
                     time  | current
                     orderedby
                     comment
                     [target="targetname"]
```

item

Name of the software package.

version

Version of the software package.

procedure

Blank string (reserved for future use).

date

Specifies the date of the installation. You can also use the keyword **current**. Format of date: YYYY-MM-DD

time

A mandatory parameter that sets a progress message for the current job container executor job.

orderedby

Specifies the User ID that ordered the installation.

comment

Your own comments.

[target="targetname"]

Specifies whether the command is applicable for the computer unit (default), or any valid or existing user profile units on the same computer. You can specify any target except the following targets:

- On the computer that you are running the command.
- From the user profile on the same computer you are running the command on.

Example

In the following example Software Delivery is told that a delivery record for software item Product Z Version 1.3 should be added. The current date and time are used for the delivery. Ordered by will list Gunnar and the comment will be Upgrade of Product Z v1.2.

```
sd_acmd addlibdeliveryrecord "Product Z" "1.3" "" current current "Gunnar" "Upgrade of Product Z v1.2"
```

Note: If you want to omit an argument, say Comment, you still have to include the empty argument in your list by typing "".

If this command is issued on a scalability server, and the software item Product Z 1.3 is present in the library of the upstream domain manager, a record will be added to the staging library folder of the scalability server. A job record will also be added to the Jobs folder of the scalability server.

AddLibRemovalRecord--Notify Software Delivery that a Software Package Was Removed from a scalability server staging library

This command adds a record notifying Software Delivery that a software package that was previously added has been removed from a scalability server staging library.

Syntax

```
AddLibRemovalRecord item
                    version
                    procedure
                    date
                    time
                    orderedby
                    comment
                    [target="targetname"]
```

item

Name of the software package.

version

Version of the software package.

procedure

Blank string (reserved for future use).

date

Specifies the date of the installation. You can also use the keyword **current**.
Format of date: YYYY-MM-DD

time

Specifies the time of the installation. You can also use the keyword **current**.
Format of time: HH:MM

orderedby

This parameter is optional and provides an account name, for example, DomainX\UserY.
Default: Current user running the SecureContainer command.

comment

Your own comments.

[target="targetname"]

Specifies whether the command is applicable for the computer unit (default), or any valid or existing user profile units on the same computer. You can specify any target except the following targets:

- On the computer that you are running the command.
- From the user profile on the same computer you are running the command on.

Example

In the following example:

```
sd_acmd addlibremovalrecord "Product W" "1.1" "" current current "Admin" "Removal of Product W v1.1"
```

Software Delivery is told that a delivery record for software item Product W Version 1.1 should be removed. The current date and time are used. Ordered by will list Admin and the comment will be "Removal of Product W v1.1."

Note: If you want to omit an argument, say Procedure, you still have to include the empty argument in your list by typing "".

If this command is issued on a scalability server, and the software item Product W 1.1 is present in the library of the upstream domain manager, and a delivery record for Product W 1.1 was present in the staging library folder of the scalability server, the delivery record will be deleted. A job record will also be added to the Jobs folder of the scalability server.

AddReinstallRecord --Software Package was Reinstalled

This command adds two records in the Jobs folder of the computer: a detection record notifying Software Delivery that a software package has been removed using a detected uninstall procedure and then a record that the software package has been installed. The install record in the Installations folder of the computer is updated with the information provided in AddReinstallRecord.

If an AddReinstallRecord is issued, and the Item or Item Procedure is missing in the software library, the installation record is automatically converted to a detection record with procedure Detected.

Syntax:

```
AddInstallRecord item
                 version
                 procedure
                 date | crurrent
                 time | current
                 orderedby
                 comment
                [target="targetname"]
```

item

Name of the software package.

version

Version of the software package.

procedure

Blank string (reserved for future use).

date

Specifies the date of the installation. You can also use the keyword **current**.
Format of date: YYYY-MM-DD

time

Specifies the time of the installation. You can also use the keyword **current**.
Format of time: HH:MM

orderedby

This parameter is optional and provides an account name, for example, DomainX\UserY.
Default: Current user running the SecureContainer command.

comment

Your own comments.

[target="targetname"]

Specifies whether the command is applicable for the computer unit (default), or any valid or existing user profile units on the same computer. You can specify any target except the following targets:

- On the computer that you are running the command.
- From the user profile on the same computer you are running the command on.

Example

In the following example Software Delivery is notified that reinstallation records for software item Product W Version 1.1, using procedure Install should be added. The current date and time are used. Ordered by will list Automatic by Job and the comment will be Required by Product X v2.0 install.

```
sd_acmd addreinstallrecord "Product W" "1.1" "Install" current current "Automatic by Job" "Required by Product X v2.0 install"
```

Note To omit an argument, include the empty argument in your list by typing "".

AddUndetectedRecord--Notify Software Delivery that a Software Package Was Removed

This commands adds a record notifying Software Delivery that a software package that was previously detected has now been removed.

Syntax

```
AddUndetectedRecord item
                    version
                    [target="targetname"]
```

item

Name of the software package.

version

Version of the software package.

[target="targetname"]

Specifies whether the command is applicable for the computer unit (default), or any valid or existing user profile units on the same computer. You can specify any target except the following targets:

- On the computer that you are running the command.
- From the user profile on the same computer you are running the command on.

Example

In the following example:

```
sd_acmd addundetectedrecord "Product V" "3.1"
```

Software Delivery is told that a record for detected software item Product V Version 3.1 should be removed.

AddUninstallRecord--Notify Software Delivery that a Software Package Was Uninstalled

This command adds a record notifying Software Delivery that a previously installed software package has been uninstalled. If there is no installation that the uninstallation record can be associated with, the result of the command is not saved.

Syntax

```
AddUninstallRecord item
                   version
                   procedure
                   date  | current
                   time  | current
                   orderedby
                   comment
                    [target="targetname"]
```

item

Name of the software package.

version

Version of the software package.

procedure

Blank string (reserved for future use).

date

Specifies the date of the installation. You can also use the keyword **current**.
Format of date: YYYY-MM-DD

time

Specifies the time of the installation. You can also use the keyword **current**.
Format of time: HH:MM

orderedby

This parameter is optional and provides an account name, for example, DomainX\UserY.
Default: Current user running the SecureContainer command.

comment

Your own comments.

[target="targetname"]

Specifies whether the command is applicable for the computer unit (default), or any valid or existing user profile units on the same computer. You can specify any target except the following targets:

- On the computer that you are running the command.
- From the user profile on the same computer you are running the command on.

Example

In the following example Software Delivery is notified that an uninstall record for software item Product Y Version 1.2, using procedure Uninstall, should be added. The current date and sda_time are used for the uninstallation. Ordered by will list Automatic by Job and the comment will be Upgraded by Product Y v2.0 install.

```
sd_acmd adduninstallrecord "Product Y" "1.2" "Uninstall" "" current current
"Automatic by Job" "Upgraded by Product Y v2.0 install"
```

Notes: If you want to omit an argument, say Installprocedure, you still have to include the empty argument in your list by typing "".

In the case previously shown, a blank string was used for the install procedure. If two installations were previously made, using Install1 and Install2 as install procedures, only one of these installations will be deleted. If however * is used instead, both installations will be deleted.

Execute Container--Executes an Software Delivery Job Container Order File

This command executes a Software Delivery job container order file.

Syntax

```
ExecuteContainer ContainerFile
                 [-o:results directory]
                 [-p:password]
                 [-t]
```

ContainerFile

Full path to the container order file.

[-o:results directory]

Full path to the results directory (optional).

[-p:password]

Password (optional).

[-t]

Time in seconds.

During execution the container order file is copied to the file *name*.cwf (for example, myorders.cwf, if the name of the container order file is myorders.cof) in the internal directory sdjexec (".\CA\DSM\Agent\units\00000001\usd\sdjexec").

When the container execution is completed, the file is renamed to *name*.crf and written to the supplied results directory (if any is given; otherwise the file is placed in the sdjexec directory).

If the $rf macro is used in the procedure parameters, the output log file is given a unique name with an extension of "res" and copied to the supplied results directory.

Notes: Empty result files are not stored on the computer.

To run the container file, the data must be secured with sd_acmd SecureContainer.

In a Windows NT/2000 environment, to be able to successfully execute a Software Delivery offline job using sd_acmd ExecuteContainer, sufficient security privileges are needed. By default, only users assigned domain or global administrative rights are authorized to run the Software Delivery offline jobs. Optionally, users who are members of the Windows NT/2000 local group SDOFFLIN or global group Domain SDOFFLIN are authorized to run Software Delivery offline jobs. The local group SDOFFLIN is created on all Windows NT/2000 computers when installing Software Delivery. administrators can create the global group Domain SDOFFLIN and add users to these groups to let users run the Software Delivery offline jobs.

In UNIX, the NIS (Network Information Service) performs similarly to domain user validation in NT. The SD NIS user group for offline jobs is also called SDOFFLIN.

Examples

The following example illustrates how to order an execution of a job container order file.

```
sd_acmd executecontainer "C:\myorders\test1\order1.cof" -o:"C:\myresults"
-p:mypassword
```

The Output Directory argument can also be added to the container order file (parameter ResultDirectory in section Container). The command line option overrides the value in the container order file.

Since a password is used to secure the container order file in the example following (SecureContainer), the (same) password parameter must also be specified when running the ExecuteContainer command. The container order file executor will internally decrypt any sensitive data before use.

-t parameter:

When the agent is busy executing a job or contacting the server the offline job executor is unable to run. The -t: parameter instructs the offline job executor to wait for a specified number of seconds before giving up and returning an error code signifying that the agent is busy. A default timeout is used if none is supplied.

```
sd_acmd executecontainer cont.cof -t:30
```

This instructs the offline job executor to wait up to 30 seconds for the agent to terminate before failing.

Encrypt--Encrypt Any String

This command encrypts any plain string that is passed as a parameter.

Syntax

```
encrypt <any-string>
```

<any-string>

Specifies a plain string.

Example

The following example returns the encrypted value of the plain string "machine1\user1", which is an account name for accessing the share:

```
sd_acmd encrypt machine1\user1
```

Similarly, the password can be entered as a plain string, and it also is encrypted.

SecureContainer--Encrypt an Software Delivery Job Container Order File

This command secures a Software Delivery job container order file by encrypting it.

Syntax

```
SecureContainer ContainerFile
                [-p:password]
```

ContainerFile

Full path to the container order file.

[-p:password]

Password (optional).

Example

The following example illustrates how all sensitive information (passwords) in the container order file order1.cof, in directory C:\myorders\test, is encrypted using the password mypassword.

```
sd_acmd securecontainer "C:\myorders\test1\order1.cof" -p:mypassword
```

WaitContainers--Waits for Jobs to Finish

This command can optionally be used after calls to ExecuteContainer in batch files to wait for all pending or running jobs to finish.

Syntax

```
WaitContainers [-t:Timeout]
               [-f:ContainerFile]
```

[-t:Timeout]

Given in seconds. Specifies the time to wait. The default value is infinite. The range is 0 to 2**31-1 (optional).

[-f:ContainerFile]

The container file name (optional). If the extension cof is left out in the file name, Software Delivery appends it.
Note: The file is assumed to reside in ".\CA\DSM\Agent\units\00000001\usd\sdjexec".

Example

The following example illustrates how to issue the command to wait for all jobs in the container order file order3.cof to finish. The timeout is 30 seconds.

```
sd_acmd waitcontainers -t:30 -f:order3.cof
```

Note: If no container file (cof file) is given, a wait for all offline jobs is initiated.

ExitCodeMsg - Will Set an Error Message for a Failed Job

This command sets an error message text for the current job container executor job.

Syntax

```
ExitCodeMsg [Error message]
```

where

Error Message

An optional parameter that determines the error message for the current job container executor job.

Example

The ExitCodeMsg command is not suitable for command line usage. It is recommended for batch and script files. For example, an executable could generate exit or return codes that indicate different error causes. By switch/case logic, an understandable error message could be generated in each such case by messages that will replace the otherwise not understandable text "Exit code xx indicates possible error".The message SDM228483 is associated with the usage of the ExitCodeMsg command.

Note: Even if the error message parameter is optional, it is not recommended to use the command without the parameter.

JobCheck--Run the Job Check

This command runs the Job Check task and activates the software delivery agent to contact the scalability server.

Three new VDI Support switches—ReportTemplateSwStateDbRecords, ReportInstanceSwStateDbRecords, and ReportAllSwStateDbRecords—have been added to extend its functionality. These three switches are only used to send job records from the instance/template databases; they do not perform offline reinstallation.

Syntax

```
JobCheck [target="targetname"]
         [update]
         [installonly]
         [/BG]
         [/wait]
         [/ReportTemplateSwStateDbRecords]
         [/ReportInstanceSwStateDbRecords]
         [/ReportAllSwStateDbRecords]
```

[target="targetname"]

Runs for the specified target, if a value is present. If no target is specified, the current context is the default if a user agent is enabled on the target computer. The parameter is optional.

[update]

Updates attribute only. The parameter is optional.

Note: The update and installonly options are mutually exclusive.

[installonly]

Does not update attribute. The parameter is optional.

Note: The update and installonly options are mutually exclusive.

[/BG]

Runs the Job Check task in background mode. The parameter is optional.

[/wait]

Ensures that sd_acmd does not return until a job check has completed. The parameter is optional.

[/ReportTemplateSwStateDbRecords]

Sends template state database records only without performing offline reinstallation. The parameter is optional.

[/ReportInstanceSwStateDbRecords]

Sends instance state database records only without performing offline reinstallation. The parameter is optional.

[/ReportAllSwStateDbRecords]

Sends all software state database records without performing offline reinstallation. The parameter is optional.

JobProgress--Will set a progress message / percentage for the Current Job

This command sets a progress message and a progress percentage for the current job.

Syntax

```
JobProgress Text
            [Percentage]
```

Text

A mandatory parameter that sets a progress message for the current job container executor job.

[Percentage]

An optional parameter that sets a progress percentage for the current job container executor job.

Example

In the following example

```
sd_acmd jobprogress "Running Job 1" 50
```

Software Delivery will present the progress message "Running Job 1" with the percentage "50" for the current job container executor job.

Note: The JobProgress command is not suitable for command line usage. It is more suitable for batch and script files. For example, executables could generate return codes that indicate different states in the whole batch or script execution scenario. A specific progress message could be generated in each such relevant state, replacing nonspecific ones.

Remove Target--Remove a Target from a Computer

This command will remove targets from computer.

Syntax:

```
sd_acmd RemoveTarget Propertystring
```

Propertystring

The property string may contain “TARGET=<target name> or TYPE=<type>.
The valid types are MACHINE, LOCAL_USER, DOMAIN_USER, PALM, WIN_CE, and NOKIA.
The TYPE= property may appear multiple times in the command if several target types are to be removed.

Examples

In the following example

```
sd_acmd removetarget TYPE=WIN_CE
```

Software Delivery will remove all records of docking devices of Windows CE type that are attached to the Win32 desktop companion.

This will clean up the record on the desktop companion, if the Windows CE device had a native WinCE agent installed after having been a docking device on the desktop companion.

Suppose now that there are two local users associated with the target computer, User1 and User2.

If you type

```
sd_acmd removetarget TARGET=T01M0234/User1
```

the local user, User 1, will be removed

If you type

```
sd_acmd removetarget TYPE=LOCAL_USER
```

all local users, User 1 and User 2, associated with the Software Delivery agent will be removed.

Remove Win32 Program

This command will run the uninstall command associated with a product registered under the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\"Product string", where the "Product String" should be entered in the <registrykey> parameter.

Syntax:

```
sd_acmd RemoveWin32Pgm RegistryKey [Propertystring]
```

RegistryKey

"Product String" of HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\"Product string"

[Propertystring]

The <Propertystring> parameter may contain the following properties:
PARAMETERS=<User added parameters> - these may be command line parameters, like for unattended processing. The value will be appended to the command line found in the “QuietUninstallString” value, and if it is absent - the “UninstallString” value.
KEYEXISTS=1 - means that only a validation is to be performed. (Using this feature “Detect Installed” and “Verify Uninstalled” procedures may easily be created. The return value is 0 if the key exists otherwise 6020).
TIMEOUT=<timeot value> - as the initial uninstall process may spawn off the real uninstall process RemoveWin32Pgm may wait the specified time in seconds for the registry key to be removed. If the key persists after process completion (+ the time out time) an error code is returned.
OUTPUT=<file name> - enables logging to file during the processing.

Example:

In the following example

```
sd_acmd removewin32pgm registrykey="Jukebox 1.0" timeput=60
```

Software Delivery should remove the Jukebox program Version 1.0 from the computer within the timeout 60 seconds.

SetDownloadMethod

This command specifies the download method for the agent.

Syntax

```
SetDownloadMethod {NOS|NONE|DTS}
```

NOS

Internal NOS

NONE

Internal NOS-less

DTS

DTS NOS-less

Note: The operation will fail if the desired download method is invalid.

For example, if DTS is not installed on the machine, "sd_acmd SetDownloadMethod DTS" will fail with message "SD_ACMD <6027>: Invalid download method".

Signal--Sends Signals to Software Delivery on Actions to Be Taken by the Software Delivery Agent

This command lets you send signals to Software Delivery on separate actions that need to be taken by the Software Delivery agent during the installation.

Syntax

```
Signal
    [reboot]
    [rerun]
    [logoff]
    [rebootae]
    [logoffae]
```

[reboot]

Causes the computer to be rebooted after execution of the current job.

[rerun]

The current job will be restarted next time as well. This command can be used with **logoff** or the **reboot** parameter.

[logoff]

Log off user after execution of the current job.

[rebootae]

Reboot computer after execution of last job in batch.

[logoffae]

Log off user after execution of last job in batch.

At least one argument is needed. Apart from that requirement, any combination of the arguments can be selected, except that rebootae and rerun cannot be used together.

Example

The following example illustrates how to signal the Software Delivery agent that the computer should be rebooted after execution of the current job and that the job should be run upon reboot. This is used in Software Delivery when, for example, executing jobs involving SXP packages. The proper logic needs to be implemented in user-written procedures to efficiently utilize this capability.

```
sd_acmd signal reboot rerun
```

ReinstallTarget--Software Reinstalled on Target

This command initiates execution of the Offline RAC process wherein any software that was deployed to the agent is restored.

Note: This command is not typically called manually as it is invoked by the VDICompose.dms integration scripts. However, if the Offline RAC process is used for a purpose other than VMware View integration, you may need to invoke it using other mechanisms.

Syntax

```
ReinstallTarget [abort]
```

[abort]

Cancels any pending reinstall target request. This parameter is optional.

Example

```
sd_acmd ReinstallTarget
```

This example performs offline reinstallation and calls the agent, which opens the instance software state database and reruns every procedure in chronological order using the current software library of the agent. The job status is reported to the domain manager and displayed in the Software Delivery Job Check dialog, similar to any standard software delivery job. In addition, the agent opens the template software state database and for each entry creates install/activate/configure records (same as generated by sd_acmd AddXXXRecord). At the end of the reinstallation process all generated records are sent up to the domain manager using the scalability server.

Note: This mechanism relies on the agent getting a new Host UUID each time it gets recomposed or refreshed. This triggers the deletion of the existing software records in the MDB for the computer (only to be replaced by new ones as soon as the Offline RAC is completed).

UserInfo - Will Set User Information for the Computer Targets

This command sets user information for the current computer targets.

Syntax

```
UserInfo [location]
         [user]
         [phone]
         [target]
         [comment]
```

[location]

An optional parameter that sets the location text for the current computer

[user]

An optional parameter that sets the user information text for the current computer

[phone]

An optional parameter that sets the phone information text for the current computer

[target]

Specifies whether the command is applicable for the computer unit (default), or any valid or existing user profile units on the same computer. You can specify any target except the following targets:

- On the computer that you are running the command.
- From the user profile on the same computer you are running the command on.

[comment]

An optional parameter that sets the comment text for the current computer

Example

In the following example

```
sd_acmd userinfo location="Room 5" user="The user" phone=123456 target=MyComputer
Comment="Building 22"
```

CA ITCM will present the specified information to a DSM Explorer user in the User Data tab in DSM Properties (Systray).

Notes: You can freely omit any argument, say Target, and you can select any combination of lowercase and uppercase letters you want.

Changing user information with this command can collide with other changes. The changes made on the computer where the command is executed are reported to the agent's manager.

If a user selects a computer target in DSM Explorer and effects an information change, that change will apply only on the server, not on the agent computer itself.

The information on the agent computer itself is not changed until a *sd_acmd userinfo* command, which initiates changes, is executed for the affected computer target. The changes made are then reported to the manager, overriding any change made using the DSM Explorer in the meantime.

A third way to set User Information is with the help of the Systray (DSM Properties / User Data tab)

Container Order File (COF) Format

The container order file, COF (.cof), is divided into the following sections:

```
Locale
        codepage=
Container
        Name=
        Password=
        Type=
        Transaction=
        OrderedBy=
        CreationTime=
        ManagerAddress=
        ResultDirectory=
        NumberOfJobs=
        NumberOfLibraries=
        BackgroundProcess=
        ShowProgress=
        Status=
        StatusMessage=
        CompletionTime=
        Secure=
        Validate=
        Hash=
        UserName=
        DomainName=
        Job_1
        Library_1
```

Locale

codepage=

This parameter is optional.
If codepage is not specified, SecureContainer (see page 28) will set it to 3.
The possible values are
1=ANSI
2=OEM
3=UTF8, or
any code page supported by the current system for example 850.

Container

Name=

Name of the container for log and output.
The name is, by default, the COF file name.

Password=

The password is used to protect the COF file.
The parameter is optional. If given, the value will be encrypted, when executing the SecureContainer command.
The parameter can also be supplied using the command SecureContainer with the optional - p: argument. In that case, the value in the command overrides the value supplied here.

Type=

This parameter denotes the container type.
Valid values are:

0	No Linkage
1	Reserved
2	Batch

Default: 0

Transaction=

This parameter denotes if transaction is used or not for the jobs in the container. It is only applicable to Software Management (SM) batch containers.
Valid values are:

0	False (not used)
1	True (used)

Default: 0

OrderedBy=

This parameter is optional and provides an account name, for example, DomainX\UserY.
Default: Current user running the SecureContainer command.

CreationTime=

(Optional) Specifies the create date and time for the container in the format YYYY-*MM-DD hh:mm*, where *YYYY* is the year, *MM* is the month, *DD* is the day, *hh* is the hour, and *mm* is the minute.

ManagerAddress=

(Optional) Provides the address to the domain manager. If no value is provided, the default value is taken from “currentmanageraddress” in the Common Configuration.
This parameter is used to replace the $csa macro.

ResultDirectory=

This parameter is optional and provides the full path to the output directory. It can be overridden by the Output Directory parameter value when running ExecuteContainer from the command line.

NumberOfJobs=

This parameter is required. It provides the number of job sections to follow in the COF file.
The default value is 1.
Note: If the value is 1, there should be only one Job section, Job_1. If the value is 2, there should be two sections, Job_1 and Job_2, and so on.

NumberOfLibraries=

This parameter is required. It provides the number of library sections to follow in the COF file.
The default value is 1.
Note: If the value is 1, there should be only one Library section, Library_1. If the value is 2, there should be two sections, Library_1 and Library_2, and so on.

BackgroundProcess=

This parameter states if jobs are to be processed in the background.
Valid values are:

-1	Use the value in the associated procedure
0	No
1	Yes

The default value is -1.
Note: Value in the associated procedure is background if the $#bg macro is used or not background if no macro is supplied.

ShowProgress=

This parameter states if the progress dialog is to be shown or hidden.
Valid values are:

0	No,
1	Yes

The default value is 1.

Status=

This parameter denotes the status of container execution.
Valid (ENU-) values are:

0	OK
1	At least one job failed.
2	Not processed.
3	COF in progress.
4	Reboot initiated.
5	Log off initiated.
6	No COF file.
7	General error reading the COF file, the file might have been tampered with.
8	Password supplied is invalid.
9	The COF file is not secured.

OUT; that is, the parameter value is output in the file *.crf in the supplied results directory.

StatusMessage=

This parameter provides status text, like a native OS message.
The value corresponds to the value of the Status parameter.
OUT; i.e., the parameter value is output in the file *.crf in the supplied results directory.

CompletionTime=

This parameter denotes the job container completion time.
It has the structure YYYY-MM-DD hh:mm (year-month-date hour:minute), for example 2001-04-09 15:34:52.
OUT; i.e., the parameter value is output in the file *.crf in the supplied results directory.

Secure=

The value of this parameter is set to 1 by executing the sd_acmd SecureContainer command.
OUT; i.e., the parameter value is output in the file *.crf in the supplied results directory.

Validate=

Encrypted value used to validate COF password. Created by sd_acmd ExecuteContainer and removed when processing is finished.

Hash=

Only valid in conjunction with -v. Forces the change to the configuration, if SDPing should fail.

UserName=

This parameter will be inserted when running ExecuteContainer. It provides the currently logged on user.

DomainName=

This parameter will be inserted when running ExecuteContainer. It provides the domain to which the current user logged on, if any.

Job_n

For details, see description of Job section (see page 44).

Library_n

For details, see description of Library section (see page 49)

Example of COF Files

The following examples of cof files are the smallest possible, using all default settings.

COF file for use of network library

```
[Container]
NumberOfJobs=1
NumberOfLibraries=1


[Library_1]
Type=1
Path=\\THE_SERVER\SDLIB


[Job_1]
SoftwareName=SuperOffice for Win2000
SoftwareVersion=7.1 ENU (I386)
ProcedureName=InstallSO
```

COF file for use of CD library

```
[Container]
NumberOfJobs=1
NumberOfLibraries=1


[Library_1]
Type=2
Path=Lib
Label=001229_1045


[Job_1]
SoftwareName=SuperOffice for Win2000
SoftwareVersion=7.1 ENU (I386)
ProcedureName=InstallSO
```

Note: A template file, template.cof, is present in the CONF directory (%SDROOT%\ASM\CONF for Windows environment, $SDROOT/asm/conf for UNIX).**Note:** There can be more than one Job and Library type section of .In such case these sections are labeled Job_2, Library_2, and so on.

Important! When you are using a text editor to create the COF you need to choose the appropriate coding when saving the COF. That is, you have to check if the current code page of your computer supports all the character you have used. If not choose the appropriate one with the codepage= parameter.

For example, if you use German umlauts, you have to choose UTF-8 (assuming codepage 3).

If codepage is not specified, SecureContainer (see page 28) will set it to 3.

COF Section Locale

codepage=

This parameter is optional.
If codepage is not specified, SecureContainer (see page 28) will set it to 3.
The possible values are
1=ANSI
2=OEM
3=UTF8, or
any code page supported by the current system for example 850.

COF Section Job

There is one (Job_1) or more sections (Job_2 and so on) with this name in the COF file. It contains the characteristics of an individual job in the job container.

Name=

This parameter provides names of the job for logging and output.
The default values are SoftwareName and ProcedureName.

Example:

```
SW:SuperOffice for Win2000 Proc:InstallSO
```

SoftwareName=

This parameter provides the name of the software package (as found in properties for the software).

Example:

```
SuperOffice for Win2000
```

SoftwareVersion=

This parameter provides the version of the package (as found in properties for the software).

Example:

```
7.1 ENU (I386)
```

ProcedureName=

This parameter provides the name of the procedure (as found in properties for the procedure) used for the job.

UserParameters=

This parameter value is used if the $up macro is present (enabling setting UserParameters here) in the associated procedure.
It is optional.
The default value is an empty string.

BootBefore=

This parameter specifies whether the target computer is to be rebooted before the job activation, and at what level this should be done.
Valid values are:

-1 Use procedure default
0 No
1 Reboot
2 Logoff

The default value is -1.
This parameter only applies to Windows NT / 2000 and Windows 9x / ME.

BootAfter=

This parameter specifies whether the target computer is to be rebooted after the job activation, and at what level this should be done.
Valid values are:

-1 Use procedure default
0 No
1 Reboot
2 Logoff
4 Reboot after the last job
8 Log off after the last job

The last two values are only of relevance if several jobs are linked together in a job container.
The default value is -1.
This parameter only applies to Windows NT / 2000 and Windows 9x / ME.

ExecutionTimeOut=

This parameter specifies the timeout value for job processing (in minutes).
Valid values are, by default, in the range 1 - 71582.
The default value is 120.

ExecutionTimeoutAction=

This parameter specifies the action to take if the timeout value above elapses.
Valid values are:

0 No action
1 Kill process

The default value is 0.

PromptRemovable=

This parameter specifies whether a prompt for media should occur, if the media is not inserted.
The parameter is only valid if the Type parameter in the associated Library section has the value 2.
Valid values are:

0 No (try next library if the device not present)
1 Yes

The default value is 0.

LogonShield=

This parameter specifies whether the logon shield should be activated.
The parameter is only applicable if the agent policy is set to “per job” (that is, the logon shield procedure Enable per job has been executed in advance on the target computer).
Valid values are:

0 No
1 Yes

The default value is 0.
This parameter only applies to Windows NT / 2000.

LibrarySearchOrder=

This parameter specifies a comma-separated list of indexes that identifies the library section search order.
This parameter is optional.
If no value is supplied, the search order is, by default, 1 if there is only a Library_1 section. It is 1,2 if there are Library_1 and Library_2 sections, and so on.

CompletionTime=

This parameter denotes the job container completion time.
It has the structure YYYY-MM-DD hh:mm (year-month-date hour:minute), for example 2001-04-09 15:34:52.
OUT; i.e., the parameter value is output in the file *.crf in the supplied results directory.

ExitCode=

This parameter stores the exit code supplied by the executable file referred to by the procedure.
OUT: the parameter value is output in the file *.crf in the supplied results directory.

ReturnCode=

This parameter stores the exit code supplied by the executable plug-in SD_JEXEC.
Valid values are:

0 OK
1 Error
2 Not executed

OUT: the parameter value is output in the file *.crf in the supplied results directory.

ErrorMessage=

This parameter stores the error text provided. It could be, for example, a native OS message.
OUT: the parameter value is output in the file *.crf in the supplied results directory.

ResultFileIn=

This parameter stores the desired name of the result file, which is generated automatically if the $rf macro is used in the associated procedure.
The path to the file will be the path that is specified by the ResultDirectory parameter in the Container section.

ResultFileOut=

This parameter stores the full path with file name to the output/log file, which is generated automatically if the $rf macro is used in the associated procedure.
OUT: the parameter value is output in the file *.crf in the supplied results directory.

Reinstall=

This parameter specifies whether the job should be a reinstall job or not.
Valid values are:
0 No
1 Yes
Default: 0

COF Section Container

There is one section with this name in the COF file. It contains the characteristics of the job container.

Name=

Password=

Type=

Transaction=

OrderedBy=

CreationTime=

ManagerAddress=

ResultDirectory=

NumberOfJobs=

NumberOfLibraries=

BackgroundProcess=

ShowProgress=

Status=

StatusMessage=

CompletionTime=

Secure=

Validate=

Hash=

UserName=

DomainName=

COF Section Library

There is one (Library_1) or more sections (Library_2 and so forth) with this name in the COF file. It contains the characteristics of the library to which the COF file refers.

Type=

Path=

MapDrive=

Label=

NOSType=

Account=

Password=

Example COF Files

The following examples of cof files are the smallest possible, using all default settings.

COF file for use of network library

```
[Container]
NumberOfJobs=1
NumberOfLibraries=1


[Library_1]
Type=1
Path=\\THE_SERVER\SDLIB


[Job_1]
SoftwareName=SuperOffice for Win2000
SoftwareVersion=7.1 ENU (I386)
ProcedureName=InstallSO
```

COF file for use of CD library

```
[Container]
NumberOfJobs=1
NumberOfLibraries=1


[Library_1]
Type=2
Path=Lib
Label=001229_1045


[Job_1]
SoftwareName=SuperOffice for Win2000
SoftwareVersion=7.1 ENU (I386)
ProcedureName=InstallSO
```

Note: A template file, template.cof, is present in the CONF directory (%SDROOT%\ASM\CONF for Windows environment, $SDROOT/asm/conf for UNIX).

Examples of COF Library Sections

```
[Library_1] ;local media
Type=0
Path=c:\lib

[Library_2] ;network
Type=1
Path=\\server\lib
Account=eunt-a01\xylzw01
Password=xyWtUG…
MapDrive=0

[Library_3] :removable
Type=2
Path=\lib
Label=”Master CD”
```

sd_acmd Return Codes

If a problem is detected by sd_acmd, a message describing the problem is printed on standard output. This message is headed "sd_acmd<xxx>: ", where *xxx* is the reported return code. The following table lists the various return codes.

-100	Too many arguments
-101	Missing arguments
-102	Syntax error
-103	Invalid date format
-104	Invalid time format
-105	Unknown command
4	Internal error occurred
110	Software name must be given
111	Software version must be given
5000	Procedure must be given
5001	Record exceeds 255 bytes
6000	Could not find Software Delivery installation
6001	Job already running
6002	Unable to find file
6003	Wrong file type
6004	Unable to create container working directory
6005	Unable to use COF file

6006	Unable to run job, Software Delivery may not be running
6007	Job already in progress, try again later
6008	Invalid password
6009	Password required to run this file
6010	Error during encryption
6011	Error loading cryptography
6012	Container already secured
6013	You do not have adequate privileges to perform this operation
6015	Timeout occurred
6016	The COF file is invalid. A section may be missing
6017	Target not found
6018	Could not trigger job check for the given target
6019	String exceeds the maximum length
6020	The requested registry key was not found
6021	Execution failed
6022	The registry key persists after execution
6023	Failed to ping given address
6027	Invalid download method (Download method not available)
6028	Cannot execute a COF file secured on a previous version of CA ITCM when operating in FIPS-only mode
6014	Unable to run job, the COF file is not secured
801	The specified target has an illegal type for this operation
1101	The specified target cannot be found
1102	The name change was unsuccessful
1103	The specified target may not be renamed at this time but a new attempt will be made the next time a JobCheck is run for this target

Chapter 3: sd_msiexe.exe

sd_msiexe.exe can be used to perform Windows Installer tasks. It is similar to Microsoft's msiexec.exe versions 1.1 and 2.0, but sd_msiexe.exe has been optimized for Software Delivery.

Note: Software Delivery macros are used in this document (for example, $msi, $iv) that are expanded to various strings during Software Delivery job processing, before processing sd_msiexe.exe. If the command is executed directly (not within an Software Delivery job), the macros may not be used, but must be replaced with real values / strings.

sd_msiexe--Perform a .msi Package Installation / Configuration / Un-installation

sd_msiexe performs a msi package installation / configuration / uninstallation.

This command has the following syntax:

```
sd_msiexe /i|x|a|jm|ju|f|t|p <package> /u <$joid> [/uv <"$iv">] [/un <"$in">]
[/q<uimode>] [/l<logmode> <LogFile>] [Property strings]
```

Note: For detailed specification, please see the msiexec.exe tool (Microsoft documentation). In the following, only the SD-specific parameters are explained.

/u--jobID

uv--SD package version

un--SD package name.

Note: /uv and /un are only required for admin and network installations using SDMSILIB.

sd_msiexe -scan--Perform a .msi package scan for the current user profile

Use sd_msiexe -scan to perform a .msi package scan for the current user profile.

The command has the following format:

`sd_msiexe -scan $cn -out:$rf (see definition on page 58)`

$cn--Should be replaced by the name of the current profile.

-out--Specifies an output file.

sd_msiexe -admdel--Delete an administrative installation for the given admin installed file.

The command sd_msiexe -admdel deletes an administrative installation for the given admin installed file.

The command has the following format:

`sd_msiexe -admdel <msipath> [-out:$rf]`

-out--Specifies an output file.

sd_msiexe -sourceupdate--Update the MSI source lists for the current profile

Use sd_msiexe -sourceupdate to update the MSI source lists for the current profile.

The command has the following format:

`sd_msiexe -sourceupdate [-out:$rf]`

-out--Specifies an output file.

sd_msiexe -updatedictionary--Update the SDMSILIB dictionary

The command sd_msiexe -updatedictionary updates the SDMSILIB dictionary.

Information from the administrative installations are added in incomplete records.

The command has the following format:

```
sd_msiexe -updatedictionary
```

sd_msiexe -removemsilibrecord--Remove a record from the SDMSILIB dictionary

Use sd_msiexe -removemsilibrecord to remove a record from the SDMSILIB dictionary.

The command has the following format:

```
sd_msiexe -removemsilibrecord ADMINPATH="$msi" | SDNAME="$in" SDVER="$iv"
```

Either the SD software name and version properties (SDNAME and SDVER) or the path property (ADMINPATH) must be added.

sd_msiexe -addmsilibrecord--Adds a record to the SDMSILIB dictionary

The command sd_msiexe -addmsilibrecord adds a record to the SDMSILIB dictionary.

This command has the following format:

```
sd_msiexe -addmsilibrecord SDNAME="$in" SDVER="$iv" ADMINPATH="$msi"
```

The correct SD SW name (SDNAME), SD SW version (SDVER) and the path to the directory (ADMINPATH) must be specified.

sd_msiexe ?|-?|/?|help--Show the usage of the sd_msiexe command

Use "?|-?|/?|help" to see the usage of the sd_msiexe command.

This command has the following format:

```
sd_msiexe ?|-?|/?|help
```

Specifies an output file.

Chapter 4: sd_sscmd--Staging Library Command Line Administration

This section describes the command-line approach (sd_sscmd command) to register new software programs and to deregister them at staging libraries.

If you want to use the command-line interface with locale parameters, you must change the console font to Lucida. This ensures that the locale characters are displayed correctly when using sd_sscmd verbose from a command prompt.

Which sd_sscmd commands are available depends whether there is a DSM manager on the system or not. When the scalability server runs on a manager machine, only a sub-set of sd_sscmd commands is available.

Important information about staging library administration commands includes the following topics:

Command Notation (see page 60)

Length Restrictions (see page 61)

Return Codes (see page 70)

Command Notification

The syntax for each command is comprised of a command and a number of associated keywords and parameters. Parameter values are case-sensitive. Command names and keywords are case-insensitive.

The delimiter between components is a space.

Parameters (and their keyword) that include spaces must be contained within quotation marks (this is not required in verbose mode). For example:

```
item="Software Test1"
```

When keywords are followed by a parameter, only one parameter value can be given.

In a batch file, for backwards compatibility both forms are supported. We recommend using quotes around the whole parameter.

Administration of a staging library is being initiated from the scalability server which hosts the staging library.

The following command syntax applies:

```
sd_sscmd
        {
          stagecheck (see page 61)
        | bulkupdate (see page 62)
        | libraryaccess {user={domainname | local\}user password=password | REMOVE}
(see page 63)
        | importmsi {mapfile=Mapfile [path=path] [move=false|true]} (see page 64)
        | addshare {SDLIBRARY | MSILIB | [UseFQDN=N|Y]} (see page 64)
        | removeshare {SDLIBRARY|MSILIB} (see page 65)
        | import {[path=path] [move=false|true]} (see page 66)
        | aregsw {path=path [reginfo=reginfo] [logfile=logfile]} (see page 66)
        | dereg {item=item version=version [logfile=logfile]} (see page 67)
        | batch {path=path [logfile=logfile]} (see page 69)
        | verbose (see page 68)
        }
```

When the scalability server runs on a manager machine, only a sub-set of commands is available:

```
sd_sscmd
        {
          stagecheck (see page 61)
        | bulkupdate (see page 62)
        | libraryaccess {user={domainname | local\}user password=password | REMOVE}
(see page 63)
        | importmsi {mapfile=Mapfile [path=path] [move=false|true]} (see page 64)
        | addshare {SDLIBRARY | MSILIB | [UseFQDN=N|Y]} (see page 64)
        | removeshare {SDLIBRARY|MSILIB} (see page 65)
        | verbose (see page 68)
        }
```

Length Restrictions

The following table summarizes the maximum length restrictions when entering values for parameters at the command line.

path 255

item 128

version 128

logfile 255

stagecheck

Valid on Windows and UNIX

The stagecheck command initiates an SD server stage check. It runs also on the manager. You can run this command whenever you want to force the scalability server to send the cached messages to the domain manager. However, if the bulk update mode is enabled and the stage check mode is disabled, the stage check functionality checks only for the unregistered units; it does not send the cached messages. The cached messages (if any) for the registered units are sent during the bulk update.

This command is independent of the specified stage check interval.

This command has the following format:

```
sd_sscmd stagecheck
```

bulkupdate

Valid on Windows and UNIX

The bulkupdate command sends a request to the scalability server to send the messages to the domain manager in bulk. Each message sent to the domain manager has specified number of submessages depending on the values provided in the configuration policies, Bulk Update: Maximum number of job results in a message and Bulk Update: Maximum units for which detection/job records will be sent in a message. If the configuration policies have not been applied to the scalability server, the default values of the configuration policies are used.

You can run the bulkupdate command whenever you want to send the cached messages in bulk. The command is independent of the bulk update interval and forces the bulk update to take place immediately. Additionally, the command does not reset the specified bulk update interval. For example, consider a scenario where you set the bulk update interval to 10 minutes, and after 5 minutes, you run the sd_sscmd bulkupdate command. In this case, the command forces the bulk update to take place immediately, but it does not reset the bulk update interval. The bulk update takes place again after 5 minutes when the actual bulk update interval of 10 minutes is elapsed.

Important! You can run this command and send the messages in bulk even when the scalability server is not running in the bulk update mode. In this case, the default value for the message size is used to send the messages in bulk.

This command has the following format:

```
sd_sscmd bulkupdate
```

Note: The command is valid for a scalability server running on the domain manager as well as for a standalone scalability server.

sd_sscmd libraryaccess

Valid on Windows and UNIX

The command libraryaccess sets the credentials that a connecting agent should use to access the SD software library. It runs also on the manager.

This command has the following format:

For Windows:

```
sd_sscmd libraryaccess {user=(domainname | local\)user password=password |
REMOVE}
```

For UNIX/Linux:

```
sd_sscmd libraryaccess {user=(domainname | local\\)user password=password |
REMOVE}
```

Note: The double backslash is required.

user

Name of the domain or "local" (current user on the local machine)\name of the user.

password

Password of the user.

REMOVE

Removes the user name and password.

Example:

```
sd_sscmd libraryaccess user=MyComputer\TheUser password=MyPassword
```

The following command removes the user name and password stored encrypted in Comstore for library access:

```
sd_sscmd libraryaccess REMOVE
```

sd_sscmd importmsi

Valid on Windows and UNIX

The command importmsi imports a MSI library from a SD 4.0 installation (used in the migration phase). It runs also on the manager.

The command has the following format:

```
importmsi {mapfile=Mapfile [path=path] [move=false | true]}
```

mapfile

mapfile is a file created during the manager migration.

path

Specifies the source path to the root of MSILIB.

If path is not given, a lookup for a SD 4 MSILIB is performed on the local machine.

move

Specifies if the files should be deleted from the original location (true) or not (false).

Default: False

sd_sscmd addshare

Valid on Windows and UNIX

The command addshare enables the shares. It runs also on the manager.

The command has the following format:

On Windows:

```
sd_sscmd addshare {SDLIBRARY | MSILIB | [UseFQDN=N|Y]}
```

On UNIX:

```
sd_sscmd addshare {SAMBA | NFS | SDLIBRARY | MSILIB | [UseFQDN=N|Y]}
```

[UseFQDN=N|Y]

Optional Parameter. Specifies that a share name includes the "Fully Qualified Domain Name" (for example, \\mycomputer.ca.com\SDMSILIB) instead of computer name only (for example, \\mycomputer\SDMSILIB).

Default: UseFQDN=Y

Note: By default, the shares SDLIBRARY$ and SDMSILIB are not created during installation (unless specified in advanced settings).

Examples

```
sd_sscmd addshare SDLIBRARY MSILIB
sd_sscmd addshare SDLIBRARY UseFQDN=Y
```

sd_sscmd removeshare

Valid on Windows and UNIX

The command removeshare removes the shares SDLIBRARY$ and SDMSILIB. It runs also on the manager.

The command has the following format:

On Windows:

```
sd_sscmd removeshare {SDLIBRARY | MSILIB}
```

On UNIX:

```
sd_sscmd removeshare {SAMBA | NFS | SDLIBRARY | MSILIB}
```

Note: By default, the shares SDLIBRARY$ and SDMSILIB are not created during installation (unless specified in advanced settings).

Example

```
sd_sscmd removeshare SDLIBRARY MSILIB
```

sd_sscmd import

Valid on Windows and UNIX

The import command imports an SD library from an SD 4.0 installation. The command is used in the migration phase.

The command has the following syntax:

```
import {[path=path] [move=false | true]}
```

path

Specifies the source path to the root of SDLIBRARY.

If path is not given, a lookup for a SD 4 SDLIBRARY is performed on the local machine.

move

Specifies if the files should be deleted from the original location (true) or not (false).

Default: False

aregsw--Automatically register software at the staging library

Valid on Windows and UNIX

Using the aregsw command, you can register a new software program that is already packaged in the standard format (for example, software previously exported from the software library).

This command has the following format:

```
aregsw {path=path [reginfo=reginfo logfile=logfile]}
```

path

The directory on the scalability server where an image of the item being registered is located. Path entries need to follow the operating system standard conventions.

The registration information subdirectory must be located in this directory.

reginfo

Path to, and name of the directory, where the registration information is to be found.

If reginfo is unspecified, sd_sscmd will look for the reginfo under the path argument.

logfile

Path to, and name of the file, where the trace information is to be written during execution of the command.

Example

The following example illustrates how to automatically register software located in the subdirectory swtest from a Windows scalability server.

The registration information found in \swtest\reginfo is used to perform the automatic registration:

```
sd_sscmd aregsw path=C:\swtest reginfo=C:\swtest\reginfo logfile=C:\test\trace1.txt
```

Note: With aregsw you should not add software to the staging library of a scalability server which is not contained in the software library of the domain manager .

This software will become visible in the "Staging Library" of the domain manager's DSM Explorer after the next job check has been performed. A job check will be started automatically when the command has successfully completed.

dereg--Deregister software items at the staging library

Valid on Windows and UNIX

This command is used to de-register or remove software from the Scalability Server SD library.

This command has the following format:

```
dereg {item=item version=version [logfile=logfile]}
```

item

The name of the item being de-registered

version

The version of the item being de-registered

logfile

Path to, and name of the file, where the trace information is to be written during execution of the command.

Example

```
dereg item="My Software Test", version=1.01, logfile=C:\test\trace2.txt
```

verbose

Verbose mode is an interactive mode that prompts for a value for each of the keywords and parameters for a particular command. The following illustrates the prompts returned when issuing a command in verbose mode.

```
D:\TEST>sd_sscmd verbose

SD Server Command Line 11.n.nnn Copyright (c) 2008 CA. All rights reserved.

Valid commands are:

        aregsw

        dereg

        quit


command :            aregsw

path  :                   D:\test

reginfo (optional) :   D:\test\reginfo

logfile (optional) :    D:\test\log1b.txt

SD_SSCMD<0>: Command successfully executed.
```

To enable verbose command support, you would first issue the command:

```
sd_sscmd verbose
```

You can choose from a list of commands. To indicate the command to execute, type the command and press Enter. The example shown illustrates how to execute the software registration command, aregsw.

After entering a command, you are prompted to supply values for each of the keywords and parameters associated with the command you specified . For example, the first parameter value you must supply for aregsw is path . After a value for path is supplied, the next prompt displays, in this case, logfile, until values for all parameters have been specified.

Note: When prompted to supply values for optional parameters, pressing Enter before entering a value will advance you to the next parameter without requiring an entry.

batch

Valid on Windows and UNIX

In batch mode, a batch file is referenced and its contents are extracted and executed. Batch files can consist of one or a series of commands.

When creating a file for batch execution, the sd_sscmd command is not listed before each command statement contained in the file. The sd_sscmd command is issued only at the onset of command execution, after which the batch file is called.

Each line in the file containing the command statements must not exceed 1023 characters.

The batch command may be used recursively. For example, in the file used as the argument for batch, you may write batch and refer to another batch file.

This command has the following format:

```
sd_sscmd batch {path = path [logfile=logfile]}
```

To enable batch mode, enter the sd_sscmd command followed by the keyword batch and the name of the file that contains the command statement or statements that you wish to execute. For example, from a Windows server you might enter:

```
sd_sscmd batch path =D:\test\regswpr.txt logfile=D:\logs\log21.txt
```

Regswpr.txt contains a command statement to register software

```
#This batch file registers one software

aregsw path=D:\test2 reginfo=D:\test2\reginfo logfile=D:\logs\log11.txt
```

All characters following the # on a line are considered part of the comment.

Ignoring Errors During Batch Processing

By default, the command-line interface stops batch file execution when an error is encountered. You can set the command-line interface options to ignore errors and continue by executing the next command contained in the batch file. To ignore errors during batch processing, enter the appropriate command.

From Windows, enter:

```
set sd_sscmd_continue = on
```

Using C-shell, enter:

```
setenv SD_SSCMD_CONTINUE ON
```

Using Bourne shell, enter:

```
SD_SSCMD_CONTINUE=ON export SD_SSCMD_CONTINUE"
```

Return Codes

All command-line administration commands generate a 0 (zero) return code when initiated successfully and a non-zero return code when an error occurs. All messages generated by command-line administration commands are prefixed with sd_sscmd.

For example:

```
sd_sscmd<4>: Command failed: Item remove error.
```

On Windows, the return code can be used to identify the reason for failure; on UNIX, the return code is either a 0, 1 or 2, indicating success, syntax error or semantic error, respectively.

The return codes listed in this section are generated when sd_sscmd is executed.

```
sd_sscmd<-1>: Syntax error.

sd_sscmd<0>: Command successfully executed.

sd_sscmd<1>: Command failed: Out of disk space.

sd_sscmd<2>: Command failed: No library path specified.

sd_sscmd<3>: Command failed: Item copy error.

sd_sscmd<4>: Command failed: Item remove error.

sd_sscmd<5>: Command failed: Volume not found.

sd_sscmd<6>: Command failed: Reg info could not be read.

sd_sscmd<7>: Command failed: Reg info not found.

sd_sscmd<8>: Command failed: Reg info not unique.

sd_sscmd<9>: Command failed: Internal error.

sd_sscmd<10>: Command failed: String conversion error.

sd_sscmd<11>: Software already registered.

sd_sscmd<12>: Software not found. Already deregistered?
```

```
sd_sscmd<13>: Too long path.

sd_sscmd<14>: Too long item name.

sd_sscmd<15>: Too long version string.

sd_sscmd<16>: Command failed: Batch file could not be opened.

sd_sscmd<17>: Timeout expired: Operation already in use

sd_sscmd<18>: Command failed: A delete operation failed

sd_sscmd<19>: Command failed: Unable to set permissions

sd_sscmd<20>: Command failed: Unable to create share

sd_sscmd<21>: Command failed: Unable to remove share

sd_sscmd<22>: Command failed: Server not running

sd_sscmd<23>: Command failed: File not found

sd_sscmd<24>: Command failed: Library not found

sd_sscmd<25>: Command failed: Previous Software Delivery installation not found

sd_sscmd<26>: Command failed: Item not found

sd_sscmd<27>: At least one operation failed
```

Index

T

U

V

W