CA IT Client Manager

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 *CLI Reference Guide* provides detailed reference information for four separate CA ITCM command line interfaces:

- cadsmcmd
- dmsweep
- dsmpgm
- dsmPush

This section contains the following topics:

Notational Conventions

Following are the notational conventions used for each command line interface.

cadsmcmd

item

cadsmcmd-keyword or constant

item

variable

|

alternative

{item}

item can be applied one or several times

[item]

item is optional

dmsweep

item

keyword or constant

<item>

variable

|

alternative

{item}

at least one of the items has to be applied

[item]

item is optional

Note: Repetition of options is not supported. Repetition of arguments is only allowed for the '/ip' option.

dsmPush

item

keyword or constant

<item>

variable

|

alternative

{item}

item can be applied one or several times

[item]

item is optional

dsmpgm

item

keyword or constant

|

alternative

{item}

item can be applied one or several times

Backslash (Windows) and Forward Slash (Linux)

The documentation uses backslashes (\) for Windows environment and forward slashes (/) for Linux environments.

Keep in mind that, when the documentation mentions a backslash, the forward slash implicitly applies to Linux.

Observe that on Linux a path beginning with / is the absolute path and a path beginning with ./ is a relative path with regard to the current directory. Relative paths are used, for example, when registering procedures using the cadsmcmd command regproc. For example, if your current directory is /a/b/c, and you state the path ./tmp/d/file1 for the file file1, the absolute path for file1 is /a/b/c/tmp/d/file1.

Chapter 2: CADSMCMD Command Line Interface

Automation is essential for the ability of managing complex networks. The CADSMCMD command line interface supports automation in the area of CA ITCM Common Object (CO), Common Configuration (CCNF), and Software Delivery (SD) including OS Installation Management (OSIM). The CLI offers shell or command prompt-based interfaces that can be utilized by shell or batch scripts for automation.

This CLI is installed with the command line component on Linux and the DSM Explorer on Windows.

Note: CADSMCMD is provided for Linux only and not other UNIX derivatives.

This section contains the following topics:

Compatibility

Due to essential changes with CA ITCM Release 12.8 the CLI is not 100% compatible with its predecessors. Some of its commands and actions have become obsolete. They are no longer described in this guide. If they are invoked, the CLI returns a warning indicating that this command or action has become obsolete. The warning is written as a message to stdout:

```
SDCMD<CMD000118>: Warning: Function is obsolete
```

The return code passed to the calling script is 0.

The information returned by some commands or actions have been changed due to the information available in Release 12.8. The list of these commands and actions is:

- targetComputer
- compGroup
- templateGroup
- jobContainer
- swLibrary
- area
- domain

If the secured object is of security class "SoftwarePackage," you have to specify the name and the version in the following format:

`«name»|«version».`

If «name» contains blanks, you have to enclose «name»|«version» in quotation marks: `"«name»|«version»".`

Example:

name testPackage and version 1.0
`soname=testPackage|1.0`

name CA ITCM Manager and version 12.0
`soname="CA ITCM Manager|12.0"`

Chapter 3: cadsmcmd Commands

This section contains the following topics:

Command Classification

The range of valid commands that the CLI offers depends on the type of manager it is connected to and the services offered by this manager.

This section contains the following topics:

Common Commands

The following commands are always available regardless to which type of manager the CLI connects (Domain or Enterprise manager) and what services this manager offers. They are Common Commands. But this does not mean that all possible actions defined with the command are available for each manager.

Command	Restricted action set
comConf	No
comgroup	Yes
servergroup	No
setdbcredentials	No

Domain

The following commands are available at a domain:

Command	required services at manager	restricted action set
activate	SD	-
aregsw	SD	-
bootserver	SD, OSIM	-
cancel	SD	-
compgroup	- SD	Yes No
configure	SD	-
dereg	SD	-
deregproc	SD	-

image	SD, OSIM	No
install	SD	-
jobcontainer	SD	No
managedcomputer	-	No
manager	SD, OSIM	-
recover	SD	-
regproc	SD	-
regsw	SD	-
renewjob	SD	-
scalabilityServer	AM	No
security	SD	-
stagingserver	SD	No
swlibrary	SD	No
swpolicy	SD	No
targetcomputer	SD/OSIM	No
templategroup	SD	No
uninstall	SD	-

Enterprise

The following commands are available at the Enterprise

Command	**required services at manager**	**restricted action set**
activate	SD	-
area	- SD	Yes No
aregsw	SD	-
cancel	SD	-
compgroup	SD	Yes Enterprise set only
configure	SD	-
copyfetched	SD	-

Command	required services at manager	restricted action set
delfetched	SD	* deprecated - use *area action=removeFetchedItem* (see definition on page 34) instead *
dereg	SD	-
deregproc	SD	-
distribution	SD	No
domain	-	No
fetch	SD	-
halt	SD	-
install	SD	-
progress	SD	-
rdereg	SD	-
rderegproc	SD	-
regproc	SD	-
regsw	SD	-
renew	SD	-
resume	SD	-
rregproc	SD	-
rregsw	SD	-
swLibrary	SD	No
swpolicy	SD	No
templategroup	SD	Enterprise set only
terminate	SD	-
uninstall	SD	-

activate—Generate an Activate Job in a Job Container

See Install command.

area—Domain Management

Actions at the enterprise:

{list (see definition on page 20) | showattr (see definition on page 23) | listmembership (see definition on page 22) | modify (see definition on page 22) |

...listdistsw (see definition on page 24) | removedistsw (see definition on page 28) | listfetcheditems (see definition on page 28) |

...listfetcheditemfiles (see definition on page 31) | removefetcheditem (see definition on page 34) |

...addtodomains (see definition on page 34) | removefromdomains (see definition on page 35)} :

Note: The terminology of CA IT Client Manager has changed in comparison with Unicenter Software Delivery 4.0. The SD "area" is equivalent to the CA ITCM "domain", while the SD "domain" is equivalent to a CA ITCM group of domains.

Nevertheless for compatibility reasons the terminology, command, and parameter names of the CLI have not changed!

To avoid any ambiguity, in the subsequent sections an area (a domain in the terminology of CA IT Client Manager) is mentioned as a "CA ITCM domain", and a domain (a domain group in the terminology of CA IT Client Manager) is mentioned as "CA ITCM domain group".

The CA ITCM domains are handled by the area command of the CLI and the CA ITCM domain groups by the domain command of the CLI.

Note: The subsequent methods are valid on enterprise manager only.

This section contains the following topics:

General Actions

This section contains the following topics:

list—List area

This action lists all CA ITCM domains. To confine the amount of information a filter can be specified.

This action has the following format:

```
area action=list
    [{filter=filter|filterfile=file_name}]
```

filter

Specifies an expression to confine the amount of information listed

The following attributes are valid to build atomic filter expressions:

Attribute	Type
Area	String
Host name	String
Registered at	String
Last Modified at	String
Comment	String
Version	String

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command.

This section contains the following topics:

Example: List All Areas

To list all areas enter the following command:

```
cadsmcmd area action=list
```

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.
Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM
--------------------------------------------------------------------------------
List of areas
--------------------------------------------------------------------------------
area 1
area 2
area4test
WWWWW01B

Number of items listed.................: 4
Number of items read...................: 4

SDCMD<A000000>: OK
```

To list the areas having the hostname starting with 'W' using filters, enter the following command:

```
cadsmcmd area action=list filter="Host name=W*"
```

The result is displayed as follows:

```
--------------------------------------------------------------------------------
List of areas
--------------------------------------------------------------------------------
WWWWW01B

Number of items listed                : 1
Number of items read                  : 4

SDCMD<A000000>: OK
```

listMembership—List Domain Groups Containing the Domain

This action lists those CA ITCM domain groups containing the specified CA ITCM domain.

This action has the following format:

```
area action=listMembership name=area_name
```

name

Specifies the name of a CA ITCM domain. Its membership in domains will be listed.

Example: List Membership in Domains

To list all the domains the area "andrea doria" is assigned to, enter the following command:

```
cadsmcmd area action=listMembership name="andrea doria"
```

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "<default manager>" as user "<default user>" ... ok.

--------------------------------------------------------------------------------
List of domains containing the area "andrea doria".
--------------------------------------------------------------------------------
test.dom

SDCMD<A000000>: OK
```

modify—Modify Attributes of Domain

This action modifies attributes of a CA ITCM domain.

This action has the following format:

```
area action=modify
    name=area_name
    [newName=new_area_name]
    [hostname=host_name]
    [comment=comment]
```

name

Specifies the name of a CA ITCM domain. It will be modified.

newName

Specifies the new name of the CA ITCM domain.

hostname

Specifies the .host name of the CA ITCM domain.

comment

Specifies a comment.

If the empty string is coded the entry is deleted.

Note: If a parameter is not coded then the related attribute remains unchanged.

Example: Modify Area Attributes

Assume the area "area4test" is to be modified. The area should be renamed to "andrea doria". The following command will change the area:

```
cadsmcmd area action=modify name=area4test newName="andrea doria"
```

showAttr—Show Attributes of Domain

This action lists the attributes of a CA ITCM domain.

This action has the following format:

```
area action=showattr name=area_name
```

name

Specifies the name of a CA ITCM domain. Its attributes are listed.

Example: Show Attributes of an Area

To list all attributes of the area "area4test" enter the following command:

```
cadsmcmd area action=showattr name=area4test
```

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off
```

```
Connecting to manager "<default manager>" as user "<default user>" ... ok.

------------------------------------------------------------------------------
Attributes of area "area4test"
------------------------------------------------------------------------------
Area.................................: area4test
Host name............................: area4test.john_doe.com
Registered at........................: 2005-04-16 11:48
Last modified at.....................: 2005-04-16 11:48
Location.............................: zenith
administrator........................: john doe
Phone................................: hein.bloed@john_doe.com
Address..............................: main st
Comment..............................:
Installation
..Version............................: 0

SDCMD<A000000>: OK
```

Distributed Software Management

This section contains the following topics:

listDistSW—List Distributed Software Products (see page 24)
removeDisSW—Remove Distributed Software Products (see page 28)

listDistSW—List Distributed Software Products

This action lists all software products distributed to a CA ITCM domain. To limit the amount of information you can specify a filter.

This action has the following format:

```
area action=listDistSW name=area_name
    [{filter=filter|filterfile=file_name}]
```

name

Specifies the name of a CA ITCM domain. For it the distributed software will be listed.

filter

Specifies an expression to confine the amount of information listed

The following attributes are valid to build atomic filter expressions:

Attribute	Type
Item name	String

Item version	String

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command.

Example:

To retrieve information about distributed software to the area "WWWWW01B" that begins with "tst" and is of Version 1.0 enter the following command:

```
cadsmcmd area action=listDistSW name=WWWWW01B filter="(Item name=tst* && Item
version=1.0)"

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "<default manager>" as user "<default user>" ... ok.

--------------------------------------------------------------------------------
List of distributed software of the area "WWWWW01B"
--------------------------------------------------------------------------------
Item name.............................: tstbase
..Item version........................: 1.0
..Distributed on......................: 2005-04-15 09:33
..Installations.......................:
....EAdm..............................: 0
....LAdm..............................: 2
....SWD...............................: 0
..Distributed procedures..............:
....Procedure name....................: inst
......Task............................: install
......Distributed on..................: 2005-04-15 09:33
....Procedure name....................: remdir
......Task............................: configure
......Distributed on..................: 2005-04-15 09:33
....Procedure name....................: setdir
......Task............................: configure
......Distributed on..................: 2005-04-15 09:33
....Procedure name....................: uninst
......Task............................: uninstall
......Distributed on..................: 2005-04-15 09:33
```

```
Item name.............................: tstkpk01
..Item version........................: 1.0
..Distributed on......................: 2005-04-15 09:33
..Installations.......................:
....EAdm..............................: 0
....LAdm..............................: 0
....SWD...............................: 0
..Distributed procedures..............:
....Procedure name....................: act
......Task............................: activate
......Distributed on..................: 2005-04-15 09:33
....Procedure name....................: actCrash
......Task............................: activate
......Distributed on..................: 2005-04-15 09:33
....Procedure name....................: conf
......Task............................: configure
......Distributed on..................: 2005-04-15 09:33
....Procedure name....................: confCrash
......Task............................: configure
......Distributed on..................: 2005-04-15 09:33
....Procedure name....................: inst
......Task............................: install
......Distributed on..................: 2005-04-15 09:33
....Procedure name....................: instCrash
......Task............................: install
......Distributed on..................: 2005-04-15 09:33
....Procedure name....................: instCrashAlways
......Task............................: install
......Distributed on..................: 2005-04-15 09:33
....Procedure name....................: uninst
......Task............................: uninstall
......Distributed on..................: 2005-04-15 09:33
....Procedure name....................: uninstCrash
......Task............................: uninstall
......Distributed on..................: 2005-04-15 09:33
Item name.............................: tstkpk02
..Item version........................: 1.0
..Distributed on......................: 2005-04-15 09:33
..Installations.......................:
....EAdm..............................: 0
....LAdm..............................: 2
....SWD...............................: 0
```

```
..Distributed procedures.............:
....Procedure name...................: act
......Task...........................: activate
......Distributed on.................: 2005-04-15 09:33
....Procedure name...................: conf
......Task...........................: configure
......Distributed on.................: 2005-04-15 09:33
....Procedure name...................: inst
......Task...........................: install
......Distributed on.................: 2005-04-15 09:33
....Procedure name...................: instCrash
......Task...........................: install
......Distributed on.................: 2005-04-15 09:33
....Procedure name...................: uninst
......Task...........................: uninstall
......Distributed on.................: 2005-04-15 09:33
Item name............................: tstkpk03
..Item version.......................: 1.0
..Distributed on.....................: 2005-04-15 09:33
..Installations......................:
....EAdm.............................: 0
....LAdm.............................: 2
....SWD..............................: 0
..Distributed procedures.............:
....Procedure name...................: act
......Task...........................: activate
......Distributed on.................: 2005-04-15 09:33
....Procedure name...................: conf
......Task...........................: configure
......Distributed on.................: 2005-04-15 09:33
....Procedure name...................: inst
......Task...........................: install
......Distributed on.................: 2005-04-15 09:33
....Procedure name...................: uninst
......Task...........................: uninstall
......Distributed on.................: 2005-04-15 09:33
Number of items listed...............: 4
Number of items read.................: 11

SDCMD<A000000>: OK
```

EAdm

Software ordered by the enterprise administrator.

LAdm

Software ordered by the domain administrator

SWD

Software detected by the Software Detector

removeDisSW—Remove Distributed Software Products

This action removes distributed software from the list of distributed software of a CA ITCM domain.

This does *not* mean that the software is removed from the related domain manager's software library.

This action has the following format:

```
area action=removeDistSW
    name=area_name
    item=item_name
    version=item_version
```

name

Specifies the name of a CA ITCM domain.

item

Specifies the name of the item. It will be removed.

version

Specifies the version of the item.

Example

To remove the software "tstkpk03" of Version 1.0 from the list of distributed software of the area "WWWWW01B" enter the following command:

```
cadsmcmd area action=removeDistSW name=WWWWW01B item=tstkpk03 version=1.0
```

Fetched Items Management

This section contains the following topics:

listFetchedItems—List Fetched Items

This action lists all fetched items of a CA ITCM domain. To limit the amount of information you can specify a filter.

This action has the following format:

```
area action=listFetchedItems name=area_name
    [{filter=filter|filterfile=file_name}]
```

name

Specifies the name of a CA ITCM domain. Its fetched items will be listed.

filter

Specifies the expression to confine the amount of information listed

The following attributes are valid to build atomic filter expressions:

Attribute	Type
Item name	String
State	[not yet] copied
Fetched at	String
Fetch path	String
Comment	String

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command.

Example:

To retrieve a list of all fetched items of area "WWWWW01B" enter the following command:

```
cadsmcmd area action=listFetchedItems name=WWWWW01B

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "<default manager>" as user "<default user>" ... ok.
```

```
--------------------------------------------------------------------------------
List of fetched items of area "WWWWW01B"
--------------------------------------------------------------------------------
Item name.............................: fetch 1 base
..State...............................: not yet copied
..Fetched at..........................: 2005-04-16 15:53
..Fetch path..........................: h:\save_sw_library\file_1\*
..Comment.............................:

Item name.............................: fetch 1 reginfo
..State...............................: not yet copied
..Fetched at..........................: 2005-04-16 15:53
..Fetch path..........................: h:\save_sw_library\file_1\reginfo\*
..Comment.............................:

Item name.............................: fetch 2 base
..State...............................: not yet copied
..Fetched at..........................: 2005-04-16 15:53
..Fetch path..........................: h:\save_sw_library\file_2\*
..Comment.............................:

Item name.............................: fetch 2 reginfo
..State...............................: not yet copied
..Fetched at..........................: 2005-04-16 15:53
..Fetch path..........................: h:\save_sw_library\file_2\reginfo\*
..Comment.............................:

Item name.............................: fetch_weu1
..State...............................: not yet copied
..Fetched at..........................: 2005-04-16 15:27
..Fetch path..........................: g:\tstool\tstbase\*
..Comment.............................:

Number of items listed................: 5
Number of items read..................: 5

SDCMD<A000000>: OK
```

To retrieve a list of all fetched items of area "WWWWW01B" having item name start with “fetch 1”, enter the following command:

```
cadsmcmd area action=listFetchedItems name=WWWWW01B filter=”Item name=fetch 1*”
```

The result is displayed as follows:

```
Item name.............................: fetch 1 base
..State...............................: not yet copied
..Fetched at..........................: 2005-04-16 15:53
..Fetch path..........................: h:\save_sw_library\file_1\*
..Comment.............................:

Item name.............................: fetch 1 reginfo
..State...............................: not yet copied
..Fetched at..........................: 2005-04-16 15:53
..Fetch path..........................: h:\save_sw_library\file_1\reginfo\*
..Comment.............................:

Number of items listed................: 2
Number of items read..................: 5
```

listFetchedItemFiles—List Fetched Item Files

This action lists the fetched files of a fetched item of the specified CA ITCM domain. To limit the amount of information, you can specify a filter.

This action has the following format:

```
area action=listFetchedItemFiles
    name=area_name
    item=item_name
    [{filter=filter | filterfile=file_name}]
    [sizeinkb]
```

name

Specifies the name of a CA ITCM domain The files of its fetched items will be listed.

Item

Specifies the name of the item. Its fetched files will be listed.

filter

Specifies an expression to confine the amount of information listed (List of fetched items)

The following attributes are valid to build atomic filter expressions:

Attribute	Type
File name	String

Attribute		Type
File size		long
File type		{directory|file}
Last modified at		String
File attributes		[R][H][S][D][A][N][T][W][X][C]
R	READONLY	
H	HIDDEN	
S	SYSTEM	
D	DIRECTORY	
A	ARCHIVE	
N	NORMAL	
T	TEMPORARY	
W	ATOMIC_WRITE	
X	XACTION_WRITE	
C	COMPRESSED	

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command.

sizeinkb

(Optional) If the sizeinkb argument is coded, the size of the fetcheditemfiles is represented in kilobytes.

Default: Bytes

Example

To retrieve information about the files of the fetched item "fetch 2 base" of the area "WWWWW01B", enter the following command:

```
cadsmcmd area action=listFetchedItemFiles name=WWWWW01B item="fetch 2 base"


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "<default manager>" as user "<default user>" ... ok.
```

```
--------------------------------------------------------------------------------
File list of fetched item "fetch 2 base" of area "WWWWW01B"
--------------------------------------------------------------------------------
File name.............................: actions.sxp
..File size...........................: 260
..File type...........................: file
..Last modified at....................: 2005-03-27 16:00
..File attibutes......................: A

File name.............................: check.sxp
..File size...........................: 166
..File type...........................: file
..Last modified at....................: 2005-03-27 16:00
..File attributes.....................: A

File name.............................: info.sxp
..File size...........................: 431
..File type...........................: file
..Last modified at....................: 2005-03-27 16:00
..File attributes.....................: A

Number of items listed................: 3
Number of items read..................: 3

SDCMD<A000000>: OK
```

To retrieve information about the file "info.sxp" of the fetched item "fetch 2 base" of the area "WWWWW01B" using filter, enter the following command:

```
cadsmcmd area action=listFetchedItemFiles name=WWWWW01B item="fetch 2 base"
filter="File name=info.sxp"
```

The output is as follows:

```
File name.............................: info.sxp
..File size...........................: 431
..File type...........................: file
..Last modified at....................: 2005-03-27 16:00
..File attributes.....................: A

Number of items listed................: 1
Number of items read..................: 3

SDCMD<A000000>: OK
```

removeFetchedItem—Remove Fetched Item

This action removes a fetched item from the CA ITCM domain's list of fetched items.

This action has the following format:

```
area action=removeFetchedItem
    name=area_name
    item=item_name
```

name

Specifies the name of a CA ITCM domain wherefrom the fetched item is removed.

Item

Specifies the name of the item that is removed.

Example:

To remove the item "fetch 2 base" from the list of fetched items of area "WWWWW01B" enter the following command:

```
cadsmcmd area action=removeFetchedItem name=WWWWW01B item="fetch 2 base"
```

Containment Management

This section contains the following topics:

addToDomains—Add Domain To Domain Groups (see page 34)
removeFromDomains—Remove Domain From Domain Groups (see page 35)

addToDomains—Add Domain To Domain Groups

This action adds a CA ITCM domain to a list of CA ITCM domain groups.

This action has the following format:

```
area action=addToDomains
    name=area_name
    {domain=domain_name}
```

name

Specifies the name of a CA ITCM domain.

It will be added to one or more CA ITCM domain groups.

domain

Specifies the name of a CA ITCM domain group.

The parameter can be coded more than once to assign a list of CA ITCM domain groups.

The CA ITCM domain groups must exist.

Example:

To add the area "WWWWW01B" to the domain "prod.dom" enter the following command:

```
cadsmcmd area action=addToDomains name=WWWWW01B domain=prod.dom
```

removeFromDomains—Remove Domain From Domain Groups

This action removes a CA ITCM domain from one or more CA ITCM domain groups.

This does not mean that the CA ITCM domain is deleted. Any containment relations other than the ones specified, remain unchanged.

This action has the following format:

```
area action=removeFromDomains
    name=area_name
    {domain=domain_name}
```

name

Specifies the name of a CA ITCM domain. It will be removed from a number of CA ITCM domain groups.

domain

Specifies the name of a CA ITCM domain group. From it the CA ITCM domain will be removed.

The parameter can be coded more than once to remove the CA ITCM domain from a list of CA ITCM domain groups.

Example:

To remove the area "WWWWW01B" from the domain "prod.dom" enter the following command:

```
cadsmcmd area action=removeFromDomains name=WWWWW01B domain=prod.dom
```

aregsw—Registering Software Automatically

The aregsw command lets you:

- Register a new software program that is already packaged in the standard format (for example, software previously exported from the software library)
- Register a new program version, which is based on a program version that is already registered in the library, and which is already packaged in the standard format.

This command has the following format:

```
aregsw  reginfo=reginfo
    path=sourcepath
    [basedonitem=baseproduct basedonversion=baseversion]
```

reginfo

Specifies the directory on the system where the registration information subdirectory is located

path

Specifies the directory on the system where an image of the item being registered is located.

basedonitem

Specifies the product name of the original item on which this new item is based.

basedonversion

Specifies the version of the base product.

Example:

The following example illustrates how to automatically register software located in subdirectory swtest2 from a Linux server. The registration information found in /test/swtest1 is used to perform the automatic registration:

```
cadsmcmd aregsw reginfo=/test/swtest1 path=/test/swtest2
```

bootserver—Managing OSIM Boot Servers

This section contains the following topics:

assignTargets—Assign Targets to Boot Server

assignTargets assigns the specified systems to the specified boot server. The systems addressed have to be registered at OSIM as un-managed or managed computers.

The syntax of the action is as follows:

```
bootServer      action=assignTargets
    name=boot_server_name
{computer=target_name|MACaddresses=MAC_address}
```

computer

Specifies the name of a OSIM managed system that is assigned to the boot server identified by the name option. If the system is already assigned to another boot server this assignment becomes obsolete and is removed. The option can be coded more than ones to specify a list.

name

Specifies the name of the boot server to which the specified list of targets will be assigned to.

MACaddresses

Specifies the MAC address of a OSIM unmanaged but registered system that is assigned to the boot server identified by the name option. If the system is already assigned to another boot server this assignment becomes obsolete and is removed. The option can be coded more than ones to specify a list.

list—List Boot Servers

This action lists the boot servers registered at the domain data base.

The syntax of the action is as follows:

```
bootServer      action=list
```

The following example shows the output of the command:

```
-----------------------------------------------------------------
List of Boot Servers
-----------------------------------------------------------------
Boot server name.....................: «name of boot server 1»
..Status.............................: {active | no response}
[..ADS server attached...............: «ADS server name»
..ADS server connection status.......: «active | not responding»]
           •
           •
           •
Boot server name.....................: «name of boot server n»
..Status.............................: {active | no response}
..[..ADS server attached.............: «ADS server name»
..ADS server connection status.......: «active | not responding»]
```

listImages—List Images at Boot Server

This action lists for a boot server registered at the domain data base the boot and OS images available at that boot server.

The syntax of the action is as follows:

```
bootServer      action=listImages
    name=boot_server_name
```

name

Name of a boot server. For this server a list of available images will be provided.

The following example shows the output of the command:

```
-----------------------------------------------------------------
List.of.Available.Images.at.<boot.server.name>
-----------------------------------------------------------------
Boot.server.name.................:.«name.of.boot.server.1»
..Image.name.....................:.«name.of.image.1»
{....Status.......................:.detected.|
....Image.type.....................:.«image.type»}
.........•
.........•
.........•
..Image.name.....................:.«name.of.image.m»
{....Status.......................:.detected.|
....Image.type.....................:.«image.type»}
```

listTargets—List Targets Registered at Boot Server

This action lists those targets registered at the specified boot server.

The syntax of the action is as follows:

```
bootServer      action=listTargets
    name=boot_server_name
    [{managedOnly | unmanagedOnly | ADSOnly}]
```

name

Specifies the name of a boot server.

managedOnly

Optional. If specified then only OSIM managed systems are reported.

unmanagedOnly

Optional. If specified then only OSIM registered but unmanaged systems are reported.

ADSonly

If specified then only ADS managed systems will be listed.

Note: If neither managedOnly nor unmanagedOnly nor ADSonly is coded then the systems will not be filtered for being OSIM managed or OSIM unmanaged.

The command provides an output list on stdout of the following format:

```
-----------------------------------------------------------------
List OSIM Targets assigned to «boot server name»
-----------------------------------------------------------------
Target name.............................: «name of target 1»
..MACAddress............................: «MAC address»
..Status................................: {OSIM managed («status» |
.......................................OSIM unmanaged | ADS managed}
            •
            •
            •
Target name.............................: «name of target n»
..MACAddress............................: «MAC address»
..Status................................: {OSIM managed («status» |
.......................................OSIM unmanaged | ADS managed}
```

cancel—Canceling a Distribution

From an enterprise manager, use the cancel command to cancel an order previously sent to domains or domain groups. The order can be canceled only if it has not yet been started, or if it has completed successfully or failed.

From a domain manager, issue the cancel command against an existing job, independent of state. However, if an active job that has been started on any target computer is canceled, it is temporarily registered at the domain manager as a failed job on those target computers on which it was installed before cancellation. The actual job status for each of these computers will be registered on the domain manager, when the job has completed.

This command has the following format:

At the enterprise:

```
cancel item=item_name
    version=version_name
    procedure=procedure_name
    task={install | activate | configure | uninstall}
    installedwith=procedure_name
    {compgrp=computer_group_name}
    {domain=domain_name | area=area_name }
    cname=[name]
```

At the domain:

```
cancel item=item_name
    version=version_name
    procedure=procedure_name
    task={install | activate | configure | uninstall}
    installedwith=procedure_name
    {compgrp=computer_group_name |
    computer=computer_name }
```

area

Specifies the name of a CA ITCM domain.

The parameter can be coded more than once to pass a list of CA ITCM domains.

cname

Specifies the the optional unique name of a job container / distribution.

If a job container / distribution with the specified name already exists an error will be reported.
If "cname" is not coded, a generic name is generated.

compgrp

Specifies the name of a computer group to be addressed by the job. The parameter can be coded more than once to assign a list of computer groups.

computer

Specifies the name of the computer.

The parameter can be coded more than once to assign a list of computers.

domain

Specifies the name of a CA ITCM domain group.

The parameter can be coded more than once to pass a list of CA ITCM domain groups.

installedWith

Specifies the name of the install procedure referred to in the job being cancelled.

This parameter is required even if the procedure being canceled is that installation procedure. In this case, enter the install procedure again with installedWith

item

Specifies the name of the item that is subject to the job being cancelled.

procedure

Specifies the name of an item procedure that is subject to the job being cancelled.

task

Specifies the type of task to be performed by the item procedure.

install

installation procedure

activate

activation procedure

configure

configuration procedure

uninstall

uninstallation procedure

version

Specifies the version of the item that is subject to the job being cancelled.

Example

In command mode, cadsmcmd is followed by the cancel command and its parameters.

```
cadsmcmd cancel item="Software TestC" version=0.46 installedwith=Install2
procedure=ConfigureC task=configure compgrp=SDCommandTest
area=N_LOCAL_SRV area=N_LOCAL_SRV_02 cname=TestCont7
```

comconf—Managing common configurations

With the command comConf you can manage common configurations. This covers the management of policies and of configurations.

This section contains the following topics:

Policies

This section contains the following topics:

createPolicy—Create a Policy

This action creates a policy.

This action has the following format:

```
comconf action=createPolicy
    [name=policy_name]
    pType=policy_type
    [comment=comment]
    [{configuration=identifier}]
```

name

Specifies the name of the policy to be created.

pType

Specifies the type of the policy to be created.

The following values are valid:

COMPUTER

Specifies a policy for computers and computer groups

IndividualComputer

Specifies a tailored computer settings

comment

Specifies a comment on or description of the policy

configuration

Specifies the UUID of a configuration the created policy will be assigned to.

If "configuration" is not coded then the policy is not assigned to any configuration.

This parameter is mandatory for policies of type "IndividualComputers".

In case of success the policy remains unsealed and the action presents the identifier of the created policy object at standard output in the following form:

```
Identifier:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

Example:

When creating a policy of name “myPolicy” of type “computer” the following command should be called:

```
cadsmcmd comConf action=createPolicy name=myPolicy pType=computer
comment=”myContainer”
```

The output provided by the command is as follows:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager: myDomMgr.ca.com
Domain: myDomMgr
Domain type: Domain
Supporting: CO CCNF USD OSIM

Identifier:1EED60D7-2CF9-11DB-A2C8-8DB9D78218A7
SDCMD<A000000>: OK
```

```
comconf action=createPolicy
    [name=policy_name]
    pType=policy_type
    [comment=comment]
    [{configuration=identifier}]
```

deletePolicy—Delete Policies

This action deletes policies.

This action has the following format:

```
comconf action=deletePolicy
    {policies=(policy_name, policy_type)}...
```

policies

Specifies the policies to be deleted by policy name and type.

Valid types:

Computer

Specifies a policy for computers and computer groups

IndividualComputer

Specifies a tailored computer settings

This parameter can be coded more than once to delete a list of policies in one call.

Example:

To delete the computer policies “myPolicy” and “mgrPolicy” just code the following:

```
cadsmcmd comConf action=deletePolicy policies=(myPolicy,computer)
policies=(mgrPolicy,computer)
```

sealPolicy—Seal Policies

This action seals policies.

This action has the following format:

```
comconf action=sealPolicy
    {policies=(policy_name, policy_type)}...
```

policies

Specifies the policies to be sealed by policy name and type.

Valid types:

Computer

Specifies a policy for computers and computer groups

IndividualComputer

Specifies a tailored computer settings

This parameter can be coded more than once to seal a list of policies in one call.

Example:

To seal the computer policies “myPolicy” and “mgrPolicy” just code the following:

```
cadsmcmd comConf action=sealPolicy policies=(myPolicy,computer)
policies=(mgrPolicy,computer)
```

showPolicy—Show Detailed Information of Policy

This action shows detailed information about the specified Policy.

This action has the following format:

```
comconf action=showPolicy
    [name=policy_name]
    [pType=policy_type]
    [DETAILED [DESC]][MEMOF]
```

name

Specifies the name of the policy of which detailed information is listed. The name might contain wildcards. In this case the information about for all the policies is listed of which their names are matching the pattern.

If not coded, the default policy will be listed.

pType

Specifies the type of the policy to be listed.

If not coded no type check is provided for the list.

The following values are valid:

Computer

Specifies a policy for computers and computer groups

IndividualComputer

Specifies a tailored computer settings

ReportedComputer

Specifies a reported computer policy

DETAILED

If coded "Parameter Sections" are listed with the policies, otherwise not.

DESC

The parameter information is listed with the parameter. If not coded the parameter information is omitted.

MEMOF

With the policies the configurations are listed the policy is a member of.

If not coded no configuration information is shown.

Example:

Basic information should be listed for the policy “enableUserProfiles”. This can be achieved by invoking:

```
cadsmcmd comConf action=showPolicy name=enableUserProfiles pType=computer
```

The information returned looks as follows:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager: myDomMgr.ca.com
Domain: myDomMgr
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
Show Policies
--------------------------------------------------------------------------------
Policy Name.............................: enableUserProfiles
..Type..................................: COMPUTER
..Comment...............................:
..Status................................: sealed

Number of policies listed: 1
```

```
SDCMD<A000000>: OK
```

Note: The name of a section is the absolute section path starting at policy's root of the format

```
 "/section_1/section_2/section_3".
```

The section name can also be seen at the Unicenter CA ITCM Explorer's Internal Name column.

When detailed information is needed then the following command should be invoked:

```
cadsmcmd comConf action=showPolicy name=enableUserProfiles
pType=computer detailed
```

The information presented might look as follows:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager: myDomMgr.ca.com
Domain: myDomMgr
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
Show Policies
--------------------------------------------------------------------------------
Policy Name.............................: enableUserProfiles
..Type..................................: COMPUTER
..Comment...............................:
..Status................................: sealed
..Parameter Sections
....Parameter Section Name..............: /itrm/agent/solutions/usd_agent
......Identifier........................: 8AC0F5E8-292F-11DB-979E-96A4044440CB
......Parameters
........Parameter Name..................: SupportedUnitTypes
..........Identifier....................: 8ACCE1AA-292F-11DB-979E-96A4044440CB
..........Value.........................: Computer ComputerUser

Number of policies listed: 1

SDCMD<A000000>: OK
```

When in addition the parameter description is needed then the following command should be invoked:

```
cadsmcmd comConf action=showPolicy name=enableUserProfiles
pType=computer detailed desc
```

This would result in the following information

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager: myDomMgr.ca.com
Domain: myDomMgr
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
Show Policies [DETAILED[desc]]
--------------------------------------------------------------------------------
Policy Name.........................: enableUserProfiles
..Type..............................: COMPUTER
..Comment...........................:
..Status............................: sealed
..Parameter Sections
....Parameter Section Name..........: /itrm/agent/solutions/usd_agent
......Identifier....................: 8AC0F5E8-292F-11DB-979E-96A4044440CB
......Parameters
........Parameter Name..............: SupportedUnitTypes
.........Identifier.................: 8ACCE1AA-292F-11DB-979E-96A4044440CB
.........Value......................: Computer ComputerUser
.........Parameter Info Name........: pi_SupportedUnitTypes
.........Attributes
...........Attribute Name...........: ppath
.............Value.................: /itrm/agent/solutions/usd_agent/suppor
.....................................+ tedunittypes
...........Attribute Name...........: desc_ja
.............Value.................:
...........Attribute Name...........: string2_ja
.............Value.................:
...........Attribute Name...........: string1_ja
.............Value.................:
```

```
...........Attribute Name.............: desc_fr
.............Value....................: Les types d'unités pris en charge par
.....................................+ le module d'extension de l'agent.
...........Attribute Name.............: string2_fr
.............Value....................: Ordinateur + profil utilisateur
...........Attribute Name.............: string1_fr
.............Value....................: Ordinateur
...........Attribute Name.............: desc_de
.............Value....................: Die vom Agent-Plugin unterstützten Ein
.....................................+ heitstypen.
...........Attribute Name.............: string2_de
.............Value....................: Computer + Benutzerprofil
...........Attribute Name.............: string1_de
.............Value....................: Computer
...........Attribute Name.............: desc_en
.............Value....................: The types of units supported by the ag
.....................................+ ent plugin.
...........Attribute Name.............: string2_en
.............Value....................: Computer + User Profile
...........Attribute Name.............: string2
.............Value....................: Computer ComputerUser
...........Attribute Name.............: string1_en
.............Value....................: Computer
...........Attribute Name.............: string1
.............Value....................: Computer
...........Attribute Name.............: type
.............Value....................: stringlist

Number of policies listed: 1

SDCMD<A000000>: OK
```

When beside the basic information additional information about the configurations the policy is being attached to is needed too then the following command should be invoked:

```
cadsmcmd comConf action=showPolicy name=enableUserProfiles
pType=computer memof
```

The information returned might look as follows:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.
```

```
Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager: myDomMgr.ca.com
Domain: myDomMgr
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
Show Policies
--------------------------------------------------------------------------------
Policy Name.........................: enableUserProfiles
..Type..............................: COMPUTER
..Comment...........................:
..Status............................: sealed
..Member of.........................: $AllComputer_update:1155623899
....Target..........................: $AllComputer

Number of policies listed: 1

SDCMD<A000000>: OK
```

unsealPolicy—Unseal Policies

This action unseals policies.

This action has the following format:

```
comconf action=unsealPolicy
    {policies=(policy_name, policy_type)}...
```

policies

Specifies the policies to be unsealed by policy name and type.

Valid types:

Computer

Specifies a policy for computers and computer groups

IndividualComputer

Specifies a tailored computer settings

This parameter can be coded more than once to unseal a list of policies in one call.

Example:

To unseal the computer policies “myPolicy” and “mgrPolicy” just code the following:

```
cadsmcmd comConf action=unsealPolicy policies=(myPolicy,computer)
policies=(mgrPolicy,computer)
```

Parameters

This section contains the following topics:

setParm—Set a Parameter and Value

This action allows to set a parameter and value.

This action has the following format:

```
comconf action=setParm
    name=policy_name,
    pType=policy_type
    parmsec=parameter_section_name
    parameter=parameter_name
    value=parameter_value
```

policy

Specifies the name of the policy the parameter belongs to.

pType

Specifies the type of the policy the parameter belongs to.

parmsec

Specifies the parameter section the parameter belongs to.

The section is absolute section path from the policy's root of format "/section_1/section_2/section_3".

The section names can either be taken from the detailed output of the showPolicy (see page 46) action or from the Internal Name column of the CA ITCM Explorer.

If the section specified is not existent at the specified policy then it is created if the section exists at the default policy too.

parameter

Specifies the name of the parameter to be set.

If the parameter is not available at the specified parameter section but at the same section at the default policy then it is created.

value

Specifies the new value of the parameter.

Example:

To enable user profiles with the SD agent the following has to be coded:

```
cadsmcmd comconf action=setParm name=myPol3 pType=Computer
parmsec=/itrm/agent/solutions/usd_agent parameter=SupportedUnitTypes
Value="Computer ComputerUser"
```

deleteParm—Delete a Parameter

This action deletes a parameter.

This action has the following format:

```
comconf action=deleteParm
    name=policy_name,pType=policy_type
    parmsec=parameter_section_name
    parameter=parameter_name
```

name

Specifies the name of the policy the parameter is removed from.

pType

Specifies the type of the policy the parameter is removed from.

parmsec

Specifies the parameter section the parameter is removed from. The section is absolute section path from the policy's root of format "/section_1/section_2/section_3".

The section names can either be taken from the detailed output of the showPolicy (see page 46) action or from the Internal Name column of the CA ITCM Explorer.

parameter

Specifies the name of the parameter to be removed.

Example:

To disable the user profile reporting from the agent again invoke:

```
cadsmcmd comconf action=deleteParm name=myPol3 pType=Computer
parmsec=/itrm/agent/solutions/usd_agent parameter=SupportedUnitTypes
```

Configurations

This section contains the following topics:

activateConfig—Activate Planned Configuration

This action activates a planned configuration for its targets. The action returns once the activation order has been scheduled, it does not wait until the activation has been completed.

This action has the following format:

```
comconf action=activateConfig
    confId=configuration_identifier
    [atTime="YYYY-MM-DD hh:mm"]
```

confId

Specifies the identifier of a planned configuration to be activated.

attime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

Note: To activate a configuration all policies attached to this configuration have to be sealed. Otherwise the activation will fail with

```
SDCMD<CMD000145>: CCNF API Request failed: 7 - Incompatible object state.
```

Example:

To activate the configuration myConfig for the 16 August, 2007, at 11:35 invoke

```
cadsmcmd comConf action=activateConfig
confId=AF2BB414-2D11-11DB-A2C8-8DB9D78218A7 attime="2007-08-16 11:35"
```

addPolicyToConfig—Add Policies to Configuration

This action adds policies to the specified configuration.

This action has the following format:

```
comconf action=addPolicyToConfig
    confId=configuration_identifier
    {{policies=(policy_name, policy_type)}
```

confId

Specifies the identifier of a configuration to which the existing specified policies are to be added.

Identifier of the configuration the existing specified policies to be added.

policies

Specifies the name and types of the policies to be assigned to the specified configuration.

The action reports success and failure of this action for the addressed configuration in the following way on stdout:

```
Configuration «configuration name», «configuration identifier»
Policy added: «policy name», «policy type», «policy identifier»
Policy failed: «policy name», «policy type», «policy identifier», «error code», «error message»
```

Example:

To add the policies “anotherPolicy” and “HideSystray” of type “computer” to the configuration “myConfig” invoke:

```
cadsmcmd comConf action=addPolicyToConfig
confId=AF2BB414-2D11-11DB-A2C8-8DB9D78218A7
policies=(anotherPolicy,computer) policies=(HideSystray,computer)
```

The information recorded by the cadsmcmd on stdout might look as follows:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager: myDomMgr.ca.com
Domain: myDomMgr
Domain type: Domain
Supporting: CO CCNF USD OSIM
```

```
Configuration myConfig, AF2BB414-2D11-11DB-A2C8-8DB9D78218A7.
Policy added : anotherPolicy, COMPUTER, 0C165644-2D16-11DB-A2C8-8DB9D78218A7
Policy added : HideSystray, COMPUTER, BBD3174E-2D06-11DB-A2C8-8DB9D78218A7
SDCMD<A000000>: OK
```

cancelConfig—Cancel Scheduled Configuration

This action cancels a scheduled configuration. The action will only succeed if the agent has not yet started with the activation of the configuration.

This action has the following format:

```
comconf action=cancelConfig
    confId=configuration_identifier
    [WAIT]
```

confId

Specifies the identifier (uuid) of the scheduled configuration to be canceled.

WAIT

If coded the action will wait for the completion of the order, otherwise it just returns after the order has been launched.

Note: If you specify WAIT, it might last some time until the action returns.

Example:

To cancel the configuration myConfig and wait for completion just invoke

```
cadsmcmd comConf action=cancelConfig
confId=AF2BB414-2D11-11DB-A2C8-8DB9D78218A7 wait
```

createConfig—Create a configuration

This action allows to create a configuration.

This action has the following format:

```
comconf action=createConfig
    name=configuration_name
    target=target_name
    ttype=target_type
    {policies=(policy_name, policy_type}...
```

name

Specifies the name of the configuration to be created.

target

Specifies the name of an existing target to be associated with the configuration.

ttype

Specifies type of the target.

The following values are valid:

Computer

The targets are computers.

This parameter is supported on domain managers only!

Group

The targets are computer groups.

If the type mismatches an error is reported.

policies

Specifies the name and types of policies which will be assigned to the configurations to be created.

Only policies of type computer could be added this way.

The created configuration is recorded to the stdout in the following format:

```
Configuration created: <<configuration name>>, <<configuration identifier>>
```

Example

To assign the policy “myPolicy” of type “computer” to a configuration “myConfig” and target it to the computer group myGroup invoke:

```
cadsmcmd comConf action=createConfig name=myConfig target=myGroup ttype=group
policies=(myPolicy,computer)
```

The information recorded at stdout might look as follows:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off
```

```
Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager: myDomMgr.ca.com
Domain: myDomMgr
Domain type: Domain
Supporting: CO CCNF USD OSIM

Configuration created: myConfig,AF2BB414-2D11-11DB-A2C8-8DB9D78218A7.
SDCMD<A000000>: OK
```

deleteConfig—Delete a Configuration

This action deletes the specified configuration.

It has the following format:

```
comconf action=deleteConfiguration
    confId=configuration_identifier}
```

confId

Specifies the the identifier of the configuration to be deleted.

Example:

To delete the configuration “obsoleteConfig” of the configuration identifier 844A8622-2D15-11DB-A2C8-8DB9D78218A7 invoke

```
cadsmcmd comConf action=deleteConfig confId=844A8622-2D15-11DB-A2C8-8DB9D78218A7
```

removePolicyFromConfig—Remove Policies from Configurations

This action removes policies from the specified configurations.

This action has the following format:

```
comconf action=removePolicyFromConfig
    confId=configuration_identifier
    {{policies=(policy_name, policy_type)}
```

confId

Specifies the identifier of a configuration from which the specified policies will be removed from.

policies

Name and types of the policies to be removed from the specified configuration.

The action reports success and failure of this action for the addressed configuration in the following way on stdout:

```
Configuration «configuration name», «configuration identifier»
Policy removed: «policy name», «policy type», «policy identifier»
Policy failed: «policy name», «policy type», «policy identifier», «error code», «error message»
```

Example

To remove the policies “anotherPolicy” and “HideSystray” of type “computer” from the configuration “myConfig” invoke:

```
cadsmcmd comConf action=removePolicyFromConfig
confId=AF2BB414-2D11-11DB-A2C8-8DB9D78218A7
policies=(anotherPolicy,computer) policies=(HideSystray,computer)
```

The information recorded by the cadsmcmd on stdout might look as follows:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager: myDomMgr.ca.com
Domain: myDomMgr
Domain type: Domain
Supporting: CO CCNF USD OSIM

Configuration myConfig, AF2BB414-2D11-11DB-A2C8-8DB9D78218A7.
Policy removed : anotherPolicy, COMPUTER,0C165644-2D16-11DB-A2C8-8DB9D78218A7
Policy removed : HideSystray, COMPUTER,BBD3174E-2D06-11DB-A2C8-8DB9D78218A7
SDCMD<A000000>: OK
```

showConfig—Show a Configuration

This action shows the contents of the specified configuration.

It has the following format:

```
comconf action=showConfig
    [{targets=(target_name, target_type)}...]
    [status=configuration_status]
    [{errors=error_code}]
    [DETAILED [DESC]]
```

targets

Names and types of targets the configurations to be listed have to address.

Valid types are "Computer" and "Group". The first is valid for domain managers only.

If the parameter is not coded, the configurations are listed regardless of what they are targeting for.

status

Specifies the status of the configurations.

Valid status are:

Planned

Planned Configuration

Scheduled

Configuration scheduled for activation

InProgress

Activation in progress

Active

Configuration is active

canceling

Current configuration is being canceled

Error

Error occurred during activation process

If not coded, the configurations will be listed independently of their status.

errors

Error occurred during activation process

If not coded, the configurations will be listed independently of any error code.

DETAILED

If coded, the configurations will be listed with policy and tailored settings information, otherwise not.

DESC

If coded, the parameter description information is listed,otherwise not.

Example:

To get a list of all configurations invoke the following command:

```
cadsmcmd comConf action=showConfig
```

The result on stdout might look as follows:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager: myDomMgr.ca.com
Domain: myDomMgr
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
Show Configuration:
--------------------------------------------------------------------------------
Configuration Name.....................: $AllComputer_update:1155623899
..Identifier...........................: A3D1923B-2C28-11DB-935A-B86845DDDD44
..Target...............................: $AllComputer
....Type...............................: GROUP
....Identifier.........................: 466FC48C-A62A-403F-A811-A1CD7081663D
..State................................: active
..Activation Date......................: 2006-08-15 08:38
```

```
Configuration Name.....................: 677-lab-pimpel_update:1155623902
..Identifier...........................: A5A4DF22-2C28-11DB-935A-B86845DDDD44
..Target...............................: 677-lab-schge2s
....Type...............................: COMPUTER
....Identifier.........................: 8348785D-9148-4ED2-B2F8-EA25082DABB3
..State................................: active
..Activation Date......................: 2006-08-15 08:38

Configuration Name.....................: pampel_update:1155623902
..Identifier...........................: A5BA544C-2C28-11DB-935A-B86845DDDD44
..Target...............................: knikl01z
....Type...............................: COMPUTER
....Identifier.........................: 89E84AE7-4DD9-470D-86AF-152264A50FD7
..State................................: active
..Activation Date......................: 2006-08-15 08:38
Configuration Name.....................: myConfig
..Identifier...........................: AF2BB414-2D11-11DB-A2C8-8DB9D78218A7
..Target...............................: myGroup
....Type...............................: GROUP
....Identifier.........................: E8E36096-3503-453A-90BD-40B3208548E2
..State................................: planned

Number of configurations listed: 4
SDCMD<A000000>: OK
```

If only the planned configuration should be listed then invoke the following

```
cadsmcmd comConf action=showConfig status=planned
```

This might result in the following output:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager: myDomMgr.ca.com
Domain: myDomMgr
Domain type: Domain
Supporting: CO CCNF USD OSIM
```

```
--------------------------------------------------------------------------------
Show Configuration:
..status......= planned
--------------------------------------------------------------------------------

Configuration Name.....................: myConfig
..Identifier...........................: AF2BB414-2D11-11DB-A2C8-8DB9D78218A7
..Target...............................: myGroup
....Type...............................: GROUP
....Identifier.........................: E8E36096-3503-453A-90BD-40B3208548E2
..State................................: planned

Number of configurations listed: 1
SDCMD<A000000>: OK
```

For getting detailed information invoke:

```
cadsmcmd comConf action=showConfig status=planned detailed
```

This might result in the following output:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager: myDomMgr.ca.com
Domain: myDomMgr
Domain type: Domain
Supporting: CO CCNF USD OSIM
```

```
--------------------------------------------------------------------------------
Show Configuration:
..status......= planned
. [DETAILED]
--------------------------------------------------------------------------------
Configuration Name.....................: myConfig
..Identifier...........................: AF2BB414-2D11-11DB-A2C8-8DB9D78218A7
..Target...............................: myGroup
....Type...............................: GROUP
....Identifier.........................: E8E36096-3503-453A-90BD-40B3208548E2
..State................................: planned
..Tailored Settings
..Policies
....Policy Name........................: myPolicy
....Type...............................: COMPUTER
....Comment............................: e.g.
....Status.............................: unsealed
....Parameter Sections
......Parameter Section Name...........: /itrm/agent/solutions/usd_agent
........Identifier.....................: 1AD0030A-2D0D-11DB-A2C8-8DB9D78218A7
........Parameters
..........Parameter Name...............: SupportedUnitTypes
............Identifier.................: D2B1604A-2D0D-11DB-A2C8-8DB9D78218A7
............Value......................: Computer ComputerUser

Number of configurations listed: 1
SDCMD<A000000>: OK
```

And for including parameter descriptions invoke

```
cadsmcmd comConf action=showConfig status=planned detailed desc
```

This might result in the following output:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off
```

```
Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager: myDomMgr.ca.com
Domain: myDomMgr
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
Show Configuration:
..status......= planned
. [DETAILED[DESC]]
--------------------------------------------------------------------------------

Configuration Name.....................: myConfig
..Identifier...........................: AF2BB414-2D11-11DB-A2C8-8DB9D78218A7
..Target...............................: myGroup
....Type...............................: GROUP
....Identifier.........................: E8E36096-3503-453A-90BD-40B3208548E2
..State................................: planned
..Tailored Settings
..Policies
....Policy Name........................: myPolicy
....Type...............................: COMPUTER
....Comment............................: e.g.
....Status.............................: unsealed
....Parameter Sections
......Parameter Section Name...........: /itrm/agent/solutions/usd_agent
........Identifier.....................: 1AD0030A-2D0D-11DB-A2C8-8DB9D78218A7
........Parameters
..........Parameter Name...............: SupportedUnitTypes
............Identifier.................: D2B1604A-2D0D-11DB-A2C8-8DB9D78218A7
............Value......................: Computer ComputerUser
............Parameter Info Name........: pi_SupportedUnitTypes
............Attributes
..............Attribute Name...........: ppath
................Value..................: /itrm/agent/solutions/usd_agent/suppor
.....................................+ tedunittypes
..............Attribute Name...........: desc_ja
................Value..................:
```

```
.............Attribute Name..........: string2_ja
.............Value.................:
.............Attribute Name..........: string1_ja
.............Value.................:
.............Attribute Name..........: desc_fr
.............Value.................: Les types d'unités pris en charge par
.....................................+ le module d'extension de l'agent.
.............Attribute Name..........: string2_fr
.............Value.................: Ordinateur + profil utilisateur
.............Attribute Name..........: string1_fr
.............Value.................: Ordinateur
.............Attribute Name..........: desc_de
.............Value.................: Die vom Agent-Plugin unterstützten Ein
.....................................+ heitstypen.
.............Attribute Name..........: string2_de
.............Value.................: Computer + Benutzerprofil
.............Attribute Name..........: string1_de
.............Value.................: Computer
.............Attribute Name..........: desc_en
.............Value.................: The types of units supported by the ag
.....................................+ ent plugin.
.............Attribute Name..........: string2_en
.............Value.................: Computer + User Profile
.............Attribute Name..........: string2
.............Value.................: Computer ComputerUser
.............Attribute Name..........: string1_en
.............Value.................: Computer
.............Attribute Name..........: string1
.............Value.................: Computer
.............Attribute Name..........: type
.............Value.................: stringlist

Number of configurations listed: 1
SDCMD<A000000>: OK
```

compgroup—Computer Group Management

A computer group (CompGroup) is a collection of target computers. All the methods of CompGroup are available on the domain and enterprise manager until stated otherwise.

This section contains the following topics:

General Group Management

This section contains the following topics:

create—Create Computer or Query Groups

The action allows you to create computer groups or query groups.

It has the following format:

```
compgroup action=create
    name=group_name
    [superGroup=name_of_supergroup]
    [comment=comment]
    [{{queryName=query |sdQuery=query }
    [engineName=engine_name
    [enginePeriod=engine_evalution_frequence]] |
    [{computer=target_computer} | addall}]
    [InheritPerms[={y|n}]]
```

addall

All SD target computers, which exist at creation time, are added to the group.

comment

Specifies a comment on the group to be created.

computer

Specifies the name of the SD target computer, user profile, or docking device to be added to the group.

The parameter can be coded more than once to add more than one object to the group.

engineName

Specifies the name of an engine that will process the group evaluation.

If not coded any engine can process the evaluation.

This parameter is only valid if queryName or sdQuery is coded.

enginePeriod

Specifies the period (in minutes) in which the engine will re-evaluate the group.

If engineName is coded but not enginePeriod, a default of 1 minute is assumed.

This parameter is only valid if engineName is coded.

inheritPerms

This parameter specifies if the group being created inherits security permissions or not.
If “InheritPerms” or “InheritPerms=y” is coded then the specified group will inherit permissions to it members and becomes a security group.
If “InheritPerms=n” is coded then the specified group will not inherit permissions to its members and therefore is no security group.
Default: InheritPerms=y .

name

Specifies the name of the computer group to be created.

queryName

Specifies the name of the query to be used for evaluating the group.

sdQuery

Specifies a legacy query to be used for evaluation of the group. The query has to be compliant with SD 4.0 SP1.

superGroup

Specifies the name of an existing computer group where the group in question will be created as a subgroup. If not coded the group will be created at the system folder "Computers and Users".

Example:

A computer group is supposed to be created:

- The group should be named LA.
- It should collect dynamically all those target computers whose names being with LA.
- The group should be updated every six hours.
- The group should initially not be a subgroup of any other group.

The following command will create such a group:

```
compGroup action=create name=LA SdQuery="Target.Name MATCHES 'LA.*'"
EnginePeriod=360
```

delete—Delete a Computer Group

This action allows you to delete computer groups.

This action has the following format:

```
compgroup action=delete name=group_name
```

name

Specifies the name of a computer group to be deleted.

Example:

To delete the group qg the following command has to be entered:

```
cadsmcmd compgroup action=delete name=qg
```

cadsmcmd.compgroup.deletewithwarn

This action allows you to delete computer groups.

This action allows you to delete computer groups with warnings for any sealed software policies or catalog groups linked to it.

This action has the following format:

```
compgroup action=deletewithwarn name=group_name
```

name

Specifies the name of a computer group to be deleted.

cadsmcmd.example.compgroup.deletewithwarm

To delete the group qg with warning, enter the following command:

```
cadsmcmd compgroup action=deletewithwarn name= qg
```

Note: If the computer group is deleted at the enterprise manager, the dependency check is done only at the enterprise manager and not on the domain manager where the group is replicated. The replication silently unlinks any software catalog group, disables any software policy, and unlinks the group before deleting the replicated group.

list—List Computer Groups

This action allows you to list all computer groups including query groups and software templates.

This action has the following format:

```
compgroup action=list
    {filter=filter |filterfile=file_name}
```

filter

Specifies the expression to confine the amount of information listed

Computer group name	string	
Group type	{Group | Query[/Template] group | Template group}	
Scope	string	
Comment	string	
Creation date	Date	
Creation time	Time	
Change date	Date	
Change time	Time	
Query	string	query groups only

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command.

Example:

Suppose a listing of all those groups, which are determined by a query and which are local, is required. The following command will generate such a list at a domain manager:

```
cadsmcmd compgroup action=list filter="(Group type=Group && Scope=local)"

--------------------------------------------------------------------------------
List of Computer Groups
Filter: (Group type=Group && Scope=local)
--------------------------------------------------------------------------------
cg_1:(Group, local)
cg_2:(Group, local)

Number of objects shown: 2
Number of objects read: 2

SDCMD<A000000>: OK
```

listMembership—List Membership of Computer Group

This action allows you to list those computer groups a computer group is contained in as a subgroup.

This action has the following format:

```
compgroup action=listMembership name=group_name
    [groupScope={global | local}]
```

name

Specifies the name of a computer group for which the memberships will be listed.

groupScope

Specifies the scope of the group specified. The following values are valid for this option:

Global

The group is a global group, created at an enterprise manager. It may be replicated to a domain manager.

Local

Specifies a local group, created at the domain manager the CADSMCMD is connected to.

If "groupScope" is not coded then the CADSMCMD first tries to locate the group as a local group. If this fails then it is tried to locate the group as a global one.

Example:

To receive a list of groups of which group "cg_2" is a member of, enter the following command:

```
cadsmcmd compgroup action=listMembership name=cg_2
```

The following information is provided:

```
------------------------------------------------------------------------------
"cg_2" is member of the following groups
------------------------------------------------------------------------------
cg_1 (Group, Operational, 2005-01-28 09:57)
cg_x (Group, Operational, 2005-01-28 10:56)
```

listSWP—List Software Policies Linked to Computer Group

This action lists all software policies that are currently linked to the specified computer group.

The syntax of the action is as follows:

```
compgroup       action=listSWP
    name=computer_group_name
    [groupScope={local|global}]
    [separator=separator_sign]
```

name

Name of the computer group for which all the software policy linked to it are listed.

groupScope

This option determines the type of computer group. The following values are valid.

«not coded»

If there is a locally created software policy of the required name then this one is taken, if not it is looked for a policy distributed from an enterprise manager. The latter is on domain managers only.

distributed

The specified software policy has to be a policy created at the enterprise manager and distributed to the addressed manager.

local

The specified software policy has to be a locally created policy.

Note: On enterprise manager all software policies have to be of type local.

separator

The separator is a character that is used for separating the different output fields in the lines of the listings. The default separator is ":" (colon).

The format of the output with default separator looks as follows:

```
----------------------------------------------------------------
List of Software Policies Linked to «group name» of Scope «group scope»
Short
----------------------------------------------------------------
«policy_1 name»:«policy_1 type»:[un]sealed
         •
         •
         •
«policy_n name»:«policy_n type»:[un]sealed
```

modify—Modify Computer Groups

The action allows you to modify computer groups.

This action has the following format:

```
compgroup action=modify
    name=group_name
    [newName=new_name]
    [comment=comment]
    [{queryName=query |sdQuery=sdquery|noQuery}]
    [engineName=engine_name
    [enginePeriod=engine_evalution_frequence]]
    [InheritPerms[={y|n}]]
```

comment

Specifies a comment.

Note: To erase an existing comment, enter the empty string ("").

engineName

Specifies the name of an engine that will process the group evaluation. If an empty string is coded then any engine can process the query

If currently no query is assigned to the group a warning is given and the parameter is ignored.

enginePeriod

Specifies in minutes the period in which the engine will re-evaluate the group. If not coded 1 min is assumed.

If currently no query is assigned to the group and no special engine to process the evaluation then a warning is given and the parameter is ignored.

inheritPerms

This optional parameter specifies if the group being modified is a security group or not.
If “InheritPerms=n” is coded then the specified group will not inherit permissions to its members and therefore is no security group any longer.
If “InheritPerms” or “InheritPerms=y” is coded then the specified group will inherit permissions to it members and therefore becomes a security group.
If the parameter is not coded then the permission inheritance settings of the group are not changed.

name

Specifies the name of a computer group to be modified.

newName

Specifies the new name of the computer group.

noQuery

A possibly coded query is deleted.

query

Indicates that the template is also a query group. The query coded has to be a USD 4.0 SP1 legacy query.

queryName

Specifies the name of the query to be used for evaluating the group.

sdQuery

Specifies the legacy query to be used for evaluation of the group. The query has to be compliant with SD 4.0 SP1.

Note: Parameters not coded will leave the related attributes unchanged.

Example:

Suppose the group "LA" has to be modified.

- The name of the group should be changed to "LA District".
- The group should now be updated every two hours.

The following command will provide these changes:

```
cadsmcmd compGroup action=modify name=LA newName="LA District" EvaluationPeriod=120
```

setDownloadMethod—Set Download Method of Computer Group

The action allows you to set the download method of one or more computer groups.

This action has the following format:

```
compGroup   action= setDownloadMethod
    [sep=separator_sign]
    {[{compgrp=computer_group_name}]
    | [{ccompgrp=computer_group_name]}
    downloadMethod={NOS|NOSLESS|DTS}
```

sep

This parameter specifies a one byte separator sign for separating a group name from its range. The separator sign should not be part of any groupName specified with the command. The default is dot (.).

compgrp

Name of a computer group addressed by this action. The parameter can be coded more than once for specifying a list of groups. The CADSMCMD tries to locate the groups specified with this parameter as local groups. If such a group does not exist then it tries to locate it as a global group.

At least one compgrp or ccompgrp parameter has to be coded with this action.

ccompgrp

Name and type of a group addressed by this action. The information passed has the following syntax:

`(group_name separator_sign {local|global})`

The separator sign is by default a dot (.) but can be changed with the “sep” parameter. If local is coded then the group is considered to be a group created on the local domain manager. If global is coded then the group is assumed to be replicated from an enterprise manager.

At least one compgrp or ccompgrp parameter has to be coded with this action.

downloadMethod

Specifies the download method to that the members of the groups will be switched. Select one of the following:

NOS	"Internal - NOS"
NOSLESS	"Internal - NOS-less"
DTS	"DTS - NOS-less"

The action only runs with domain managers having SD installed.

showAttr—Show Attributes of Computer Groups

This action allows you to list the attributes of computer groups.

This action has the following format:

```
compgroup action=showattr
    name=group_name
    [groupScope={global | local}]
```

name

Specifies the name of a computer group of which the attributes are listed.

groupScope

Specifies the scope of the group specified. The following values are valid for this option:

Global

The group is a global group, created at an enterprise manager. It may be replicated to a domain manager.

Local

Specifies a local group, created at the domain manager the CADSMCMD is connected to.

If "groupScope" is not coded then the CADSMCMD first tries to locate the group as a local group. If this fails then it is tried to locate the group as a global one.

Note: For a template group the method is mapped to the templategroup action showattr (see definition on page 461).

Example:

To show the attributes of a computer group, launch the following command:

```
cadsmcmd compgroup action=showAttr name=cg
```

Output

Standard Groups

For a standard computer group - for example, cg_1, the output is as follows:

```
-------------------------------------------------------------------------------
Attributes of the computer group "cg_1"
-------------------------------------------------------------------------------
Computer group name...................: cg_1
Group type............................: Group
Scope.................................: Local
State.................................: Operational
Comment...............................:

Security type.........................:  security group
Creation date.........................: 2005-01-28
Creation time.........................: 09:57
Change date...........................: 2005-01-28
Change time...........................: 09:57
```

Query Groups

For a query group - for example, "qg", the output is:

```
-------------------------------------------------------------------------------
Attributes of the computer group "qg"
-------------------------------------------------------------------------------
Computer group name...................: qg
Group type............................: Query group
Scope.................................: Local
State.................................: Operational
Comment...............................:
Security type.........................:  security group
Creation date.........................: 2005-01-21
Creation time.........................: 16:06
Change date...........................: 2005-01-21
Change time...........................: 16:06
Query.................................: USD_qg [01/21/05 16:06:56.588]
Engine name...........................: <All Engines>
Engine period (minutes)...............: 60
```

Software Templates

Note: For a template group the method is mapped to the templategroup action showattr (see definition on page 461).

Example:

To show the attributes of a computer group, launch the following command:

```
cadsmcmd compgroup action=showAttr name=cg
```

Output

Standard Groups

For a standard computer group - for example, cg_1, the output is as follows:

```
--------------------------------------------------------------------------------
Attributes of the computer group "cg_1"
--------------------------------------------------------------------------------
Computer group name...................: cg_1
Group type............................: Group
Scope.................................: Local
State.................................: Operational
Comment...............................:
Creation date.........................: 2005-01-28
Creation time.........................: 09:57
Change date...........................: 2005-01-28
Change time...........................: 09:57
```

Query Groups

For a query group - for example, "qg", the output is:

```
--------------------------------------------------------------------------------
Attributes of the computer group "qg"
--------------------------------------------------------------------------------
Computer group name...................: qg
Group type............................: Query group
Scope.................................: Local
State.................................: Operational
Comment...............................:
Creation date.........................: 2005-01-21
Creation time.........................: 16:06
Change date...........................: 2005-01-21
Change time...........................: 16:06
Query.................................: USD_qg [01/21/05 16:06:56.588]
Engine name...........................: <All Engines>
Engine period (minutes)...............: 60
```

Software Templates

Note: For a template group the method is mapped to the templategroup action showattr (see definition on page 461).

Membership Management

The subsequent methods of this chapter run on domain manager only.

This section contains the following topics:

add—Add Computer to Computer Group

This action allows you to add computers to the computer group.

This action has the following format:

```
compgroup action=add
    name=group_name
    {{computer=computer_name} | addall}
```

addall

All SD target computers, which exist at creation time, are added to the group.

computer

This parameter specifies the name of an SD target computer, user profile, or docking device to be added to the group.
The parameter can be coded more than once for adding more than one object to the group.

name

Specifies the name of a computer group to which the computers will be added.

Example:

To add the machines HOHOH01A and KEKEK01A to the group cg_1 enter the following command:

```
cadsmcmd compgroup action=add name=cg_1 computer=HOHOH01A computer=KEKEK01A
```

Note: If cg_1 is associated with a query this operation will fail. The list of machines assigned to a query group, can only be modified by changing or reevaluating the query.

listmem—List Members of Computer Group

The action allows the lististing of members of a group.

It has the following format:

```
compgroup action=listMem
    name=group_name
    [groupScope={global | local}]
```

name

Specifies the name of a computer group

groupScope

Specifies the scope of the computer group specified. The following values are valid for this option:

Global

The group is a global group, created at an enterprise manager. It may be replicated to a domain manager.

Local

Local scope (domain)

If "groupScope" is not coded then the CADSMCMD first tries to locate the group as a local group. If this fails then it is tried to locate the group as a global one.

The following output list will be provided:

```
--------------------------------------------------------------------------------
Members of computer group "(<group name>, <group scope>)"
--------------------------------------------------------------------------------
<group member>
<group member>
<group member>
...
```

Example:

To receive the list of members in group "cg_1", enter the following command:

```
cadsmcmd compgroup action=listMem name=cg_1
```

The output will be as follows:

```
--------------------------------------------------------------------------------
Members of computer group "(cg_1, local)"
--------------------------------------------------------------------------------
GRRGGRRGC
GRGRGRGRU02
HOHOH01A
HOHOH01B
KEKEK01A
KUKUK01A
KUKUK01B
WAUWA01
```

remove—Remove Computer from Computer Group

If you are connected to a domain manager, you can remove a member from a computer group, by deleting the computer or by unlinking it.

This action has the following format:

```
compgroup action=remove
    name=group_name
    {computer=computer_name}
```

name

Specifies the name of a computer group from which one or more members are to be removed.

computer

This parameter specifies the name of an SD target computer, user profile, or docking device to be removed from the group.
The parameter can be coded more than once for removing more than one object from the group.

Example:

To remove the machines HOHOH01A and KEKEK01A from the group cg_1 enter the following command:

```
cadsmcmd compgroup action=remove name=cg_1 computer=HOHOH01A computer=KEKEK01A
```

Note: If cg_1 is associated with a query this operation will fail. The list of machines assigned to a query group can only be modified by changing or reevaluating the query.

listFailedJobs—List Failed Jobs of a Computer Group

This action lists detailed information about failed jobs that are associated with the members of this group.

This action is valid for domain managers only.

This action has the following format:

```
compgroup action=listFailedJobs
    name=computer_group_name
    [groupScope={global | local}]
```

Name

Specifies the name of an existing computer group. Information about existing failed jobs associated with the group members is listed.

The name might contain database wildcards, for example,
“%” for an arbitrary even empty sequence of characters
”_” for an arbitrary single character.

groupScope

Specifies the scope of the group specified. The following values are valid for this option:

Global

The group is a global group, created at an enterprise manager. It may be replicated to a domain manager.

Local

Specifies a local group, created at the domain manager the CADSMCMD is connected to.

If groupScope is not coded then the groups are not checked for being a local or global group.

The information returned is formatted in the same way as for the action listJobs (see definition on page 83) of compgroup.

listJobs—List Information about Existing Jobs Associated with the Members of the Group

This action lists detailed information about existing jobs that are associated with the members of this group.

This action is valid for domain managers only.

This action has the following format:

```
compgroup action=listJobs
    name=group_name
    [groupScope={global | local}]
```

Name

Name of an existing computer group. Information about existing failed jobs associated with the group members is listed.

The name might contain database wildcards, for example,
“%” for an arbitrary even empty sequence of characters
”_” for an arbitrary single character.

groupScope

Specifies the scope of the group specified. The following values are valid for this option:

Global

The group is a global group, created at an enterprise manager. It may be replicated to a domain manager.

Local

Specifies a local group, created at the domain manager the CADSMCMD is connected to.

If groupScope is not coded then the groups are not checked for being a local or global group.

The information retrieved is presented as follows:

```
--------------------------------------------------------------------------------
List of jobs associated with group group_name

--------------------------------------------------------------------------------
Computer group name......: group_name
..Target................: target computer name
....Job name............: job name
....Job identifier.......: computer job identifier
....Item name............: item name
....Item version.........: item version
....Procedure name.......: procedure name
....Task.................: task
....User parameters......: user parameters
....Job state............: job status
....Error code...........: error code
....Error message........: error message
....Creation date........: yyyy-mm-dd
....Creation time........: hh:mm
....Activation date......: yyyy-mm-dd
....Activation time......: hh:mm
....Completion date......: yyyy-mm-dd
....Completion time......: hh:mm
```

Hierarchy Management

This section contains the following topics:

addSubgroups—Add Subgroups to Computer Group

This action allows you to add subgroups to a group.

This action has the following format:

```
compgroup action=addSubgroups
    [name=group_name]
    {compgrp=group_name}
```

name

Specifies the name of the computer group the specified computer groups of "compgrp" will be added to.

The group should not be associated with a query.

If this parameter is not coded, the subgroups are copied to the container "Computers and Users".

compgrp

Specifies the name of the computer group to be added.

The parameter can be coded more than once to add the subgroup to a list of computer groups.

Note: The group to which the computer groups are added should not be a query group.

Example:

Suppose the groups named "LA admin," "LA sales," and "LA rd," should be added to the group "LA District". The following command will assign these groups as subgroups to the supergroup "LA District":

```
cadsmcmd compGroup action=addSubgroups name="LA District" compgrp="LA admin"
compgrp="LA sales" compgrp="LA rd"
```

By the way, the groups "LA admin," "LA sales," and "LA rd," have to be defined at the time of command invocation, otherwise, an error is reported for the missing subgroup.

listSubgroups—List Subgroups

This action allows you to list the subgroups of a group.

This action has the following format:

```
compgroup action=listSubgroups
    [name=group_name]
    [groupScope={global | local}]
```

name

Specifies the name of a computer group of which the subgroups will be listed.

If this parameter is not coded the computer groups of the folder "Computers and Users" are listed.

groupScope

Specifies the scope of the group specified. The following values are valid for this option:

Global

The group is a global group, created at an enterprise manager. It may be replicated to a domain manager.

Local

Specifies a local group, created at the domain manager the CADSMCMD is connected to.

If "groupScope" is not coded then the CADSMCMD first tries to locate the group as a local group. If this fails then it is tried to locate the group as a global one.

Example:

To list all subgroups associated with the group "LA District" code the following:

```
cadsmcmd compGroup action=listSubgroups name="LA District"
```

The output list has the following format:

```
--------------------------------------------------------------------------------
List of Subgroups of Computer Group "LA District"
--------------------------------------------------------------------------------
LA admin:(Group, Local)
LA sales:(Query group, Local)
LA rd:(Template group, Local, 1)
LA pr:(Query/Template Group, Local, 1)
```

The figures in the template group line indicate the number of templates associated with the groups.

removeSubgroups—Remove Subgroups

This action allows you to remove subgroups from a computer group.

This action has the following format:

```
compgroup action=removeSubgroups
    [name=group_name]
    {compgrp=group_name}
```

name

Specifies the name of a computer group the specified computer groups will be removed from. The group should not be associated with a query. If this parameter is not coded then the subgroups are removed from the folder "Computer and User Groups".

compgrp

Specifies the name of a computer group to be removed.

The parameter can be coded more than once to remove a list of computer groups.

Notes: "Remove" does not mean that any computer group of the "compgrp" parameter is deleted, it is only unlinked from the supergroup specified with the name parameter.

The group specified by the name parameter should not be a query group.

Example:

Suppose that the groups named "LA admin," "LA sales," and "LA rd," should be removed from the group "LA District" again. The following command will remove these groups as subgroups from the supergroup "LA District":

```
cadsmcmd compGroup action=removeSubgroups name="LA District" compgrp="LA admin"
compgrp="LA sales" compgrp="LA rd"
```

The groups "LA admin," "LA sales," and "LA rd," are not deleted from the database by the command; they are only removed from the group "LA District". Any other containment relations beside the ones previously shown are still present.

The command will report an error if the containment relation to be broken is the only one for one of the subgroups to be removed. For example, "LA admin" is only mentioned as a subgroup of "LA District" while the other subgroups are also mentioned in the folder "Computer and User Groups" or in other groups. Then "LA admin" is not removed, but the others are. The message "A001513 – Illegal Computer Group operation. Group must have at least one super group". is launched. The only way to remove "LA admin" is to delete it (cadsmcmd compGroup action=delete name="LA admin") or assign it as a subgroup to another group and then remove it from "LA District".

configure—Generate a Configure Job in a Job Container

See Install command.

copyfetched—Copying Fetching Items

The copyfetched command is valid only when issued from the enterprise manager.

Copyfetched is used to get access to files that have been fetched from the domain manager to the enterprise manager. Such fetched files are stored in a fetch area of the enterprise manager. Before the files can be accessed, they need to be copied to another location on the enterprise manager.

This command has the following format:

```
copyfetched area=area_name
    item=fetched_item_name
    path=enterprise_manager_path_name
```

area

Specifies the name of a CA ITCM domain where the files were fetched.

item

Specifies the name of the fetched item. This item name was used by the previous Fetch operation that retrieved the item.

path

Specifies the path to the items to be fetched., where the fetched item is to be placed.

The path is created if it does not already exist.

Example:

In the following example, copyfetched is issued from a Windows enterprise manager. "Tested Software" has been fetched from the "Charolette" area and copied to the directory, "C:\TEST\FETCHED1" on the enterprise manager.

```
cadsmcmd copyfetched area=CHAROLETTE item=" Tested Software"path=c:\test\fetched1
```

dereg—Deregistering Items

This command is used to deregister or remove software from the enterprise or domain library.

This command has the following format:

```
dereg  item=item_name  version=item_version
```

item

Specifies the name of the item being deregistered

version

Specifies the version of the item being deregistered.

deregproc—Deregistering Procedures

Use this command to deregister (remove from the enterprise or domain library) an item procedure associated with a software program. The item procedure cannot be an embedded item procedure when the product is sealed already.

This command has the following format:

```
deregproc  item=item_name
    version=item_version
    procedure=procedure_name
```

item

Specifies the name of the item to which this procedure belongs.

version

Specifies the version of the item to which this procedure belongs.

procedure

Specifies the name of a procedure. It will be deregistered.

distribution—Distribution Command

This action is valid only for enterprise managers.

This section contains the following topics:

delete—Delete Distribution

The distribution in question will be deleted.

This action has the following format:

```
distribution action=delete
    name=distribution_name
```

name

Specifies the name of the distribution to be deleted.

list—List Distributions

This action shows the list of distributions.The list contains only those distributions that the invoking user is authorized to list.The list shows the name of the distribution and its status (in round brackets).

This action has the following format:

```
distribution action=list
```

The following information is shown:

```
--------------------------------------------------------------------------------
List of distributions
--------------------------------------------------------------------------------
send prod (ok)
send instconf (ok)
remove prod (building)
remote reg (ok)
multi (waiting)
--------------------------------------------------------------------------------
Number of distributions found: 5
Number of distributions shown: 5
```

The list shows the name of the distribution and its status in parentheses.

setDistPriority—Set Distribution Priority

This action changes the priority of the distribution regardless whether the specified distribution is sealed or not.

This action has the following format:

```
distribution action=setDistPriority
    name=distribution_name
    [distPriority={1,...,10}]
```

name

Specifies the name of the distribution of which the priority will be changed.

distPriority

Specifies the new priority that will be assigned to the distribution.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

showAttr—Show Attributes of a Distribution

This action shows the attributes of a distribution.

This action has the following format:

```
distribution action=showattr name=distribution_name
```

name

Specifies the name of the distribution for which the attributes will be shown.

The following information is shown:

```
Please, enter one of the commands above.............: distribution
{list | showattr | showorderdetails |
...delete | setdistpriority}.......................: showattr
distribution name=<name>............................: DIST_ins_xxx003_def

-------------------------------------------------------------------------------
Show attributes of the distribution "DIST_ins_xxx003_def"
-------------------------------------------------------------------------------
Distribution priority..................: 5
Distribution status....................: error
Permission mask........................: 13
.....................................delete allowed
```

```
.....................................renew allowed
.....................................resume allowed
Creation date.......................: 2006-08-16
Creation time.......................: 13:50
Distribution  date..................: 2006-08-16
Distribution time...................: 13:50
Halt date...........................: 2006-08-16
Halt time...........................: 19:50
Number of sites.....................: 1
.....................................0 - waiting
.....................................0 - active
.....................................1 - failed
.....................................0 - ok
..Area..............................: xxx-yyy-zzz
....Location........................:
....Data sent in per cent...........: 100
....Distribution status.............: ok
....Order status....................: error
....Order number failed.............: 1
....Order error code................: 301820
....Order error message.............: Distribution Status Error: Order no: 1
.....................................+..and the following orders could not
.....................................+ be executed. Reason: Could not find
.....................................+ Item related to Item Procedure, or I
.....................................+ tem was registered locally.
Orders..............................: 1
..Order name:.......................: Reg Job Cnr:DIST_ins_xxx003_def
....Order number....................: 1
....Job container priority..........: 5
....Jobs............................: 1
......Job name......................: tstxxx_3_es 1.0:inst
........Job order number............: 0
........Job state...................: WAITING

SDCMD<A000000>: OK
```

showOrderDetails—Show Order Details of a Distribution

This action shows order details of a distribution.

This action has the following format:

```
distribution action=showorderdetails
    name=distribution_name
    ordername=order_name
```

name

Specifies the name of the distribution for which the order details will be shown.

ordername

Specifies the name of the order.

The following information is shown:

```
Please, enter one of the commands above.............: distribution
{list | showattr | showorderdetails |
...delete | setdistpriority}........................: showorderDetails
distribution name=<name>............................: kkkkk01k [16.08.2006
08:04:40]
ordername=<name>....................................: Reg Job Cnr:kkkkk01k
[16.08.2006 08:04:40]

--------------------------------------------------------------------------------
Details about order "Reg Job Cnr:kkkkk01k [16.08.2006 08:04:40]"
........of distribution "kkkkk01k [16.08.2006 08:04:40]"
--------------------------------------------------------------------------------
..Order name:.........................: Reg Job Cnr:kkkkk01k [16.08.2006 08:04
.......................................+ :40]
....Order number......................: 1
....Order properties..................: 1
.......................................No linkage
....Job container priority............: 5
....Jobs..............................: 1
......Job name........................: tstbase 1.0: inst
........Item name.....................: tstbase
........Item version..................: 1.0
........Item type.....................: SOFTWARE
........Procedure name................: inst
........Job state.....................: WAITING
.......................................0 - waiting
.......................................0 - active
.......................................0 - ok
.......................................0 - failed
........Job order number..............: 0
........Creation date.................: 2006-08-16
........Creation time.................: 08:05
........Delivery date.................: 2006-08-30
........Delivery time.................: 08:04
........Activation date...............: 2006-08-31
........Activation time...............: 08:04
```

```
........Delivery calendar..............:
........Timeout <days/hours>...........: 7 / 0
........Job startup....................: local time/exact
........Job permission.................: 2
.....................................cancel allowed
........Preaction.....................: no action
........Postaction....................: no action
........Job operation mask............: 4
.....................................calendar controlled
........User parameters................:
........Job execution mode............: 1
.....................................execute
........Prompt timeout <days/hours>....: 1 / 0
........Job task......................: 1 - install

SDCMD<A000000>: OK
```

Note: The information shown depends on the type of the order.

domain—Domain Group Management

The methods of this section are valid for enterprise manager only.

Note: The terminology of CA IT Client Manager has changed in comparison with Unicenter Software Delivery 4.0. The SD "area" is equivalent to the CA ITCM "domain", while the SD "domain" is equivalent to a CA ITCM group of domains.

Nevertheless for compatibility reasons the terminology, command, and parameter names of the CLI have not changed!

To avoid any ambiguity, in the subsequent sections an area (a domain in the terminology of CA IT Client Manager) is mentioned as a "CA ITCM domain", and a domain (a domain group in the terminology of CA IT Client Manager) is mentioned as "CA ITCM domain group".

The CA ITCM domains are handled by the area command of the CLI and the CA ITCM domain groups by the domain command of the CLI.

This section contains the following topics:

add—Add Domains to Domain Groups

This action adds one, several or all CA ITCM domains to a specified CA ITCM domain groups.

This action has the following format:

```
domain  action=add
    name=domain_name
    {area=area_name | addall}
```

name

Specifies the name of a CA ITCM domain group to which the CA ITCM domains specified by the "area" parameter will be added.

area

Specifies the name of a CA ITCM domain to be added to the specified CA ITCM domain group.

The parameter can be coded more than once to add a list of CA ITCM domains to the group.

addall

Adds all registered CA ITCM domains to the specified CA ITCM domain group.

Example:

To add the areas "area 1" and "area 2" to the domain "dom f," the following command should be entered:

```
cadsmcmd domain action=add name="dom f" area="area 1" area="area 2"
```

To add all the areas defined to the domain "dom f," the following command should be entered:

```
cadsmcmd domain action=add name="dom f" addall
```

addSubDomains—Add Domain Groups to Domain Groups

This action adds CA ITCM domain groups as CA ITCM domain groups to the specified CA ITCM domain group.

This action has the following format:

```
domain action=addSubdomains [name=domainname] {domain=domainname}
```

name

Specifies the name of a CA ITCM domain group (which must already exist) to which the CA ITCM domain groups will be added. The domain must exist.

If not coded, the CA ITCM domain groups are added at the top level of the folder "Domains".

domain

Specifies the name of a CA ITCM domain group which must exist and will become a CA ITCM domain subgroup of the CA ITCM domain group specified with the "name" parameter.

The parameter can be coded more than once to add a list of CA ITCM domain groups.

Example:

Assume the domain "dom e" should also appear at the top level under the folder "Domains". The following command will link the domain to this level:

```
cadsmcmd domain action=addSubdomains domain="dom e"
```

Assume the domains "dom f" and "dom g" are to be added as subdomains to domain "dom e". The following command will link these domains to "dom e" :

```
cadsmcmd domain action=addSubdomains name="dom e" domain="dom f" domain="dom g"
```

create—Create Domain Group

This action creates a new CA ITCM domain group. The CA ITCM domain group can be populated with CA ITCM domains during creation.

This action has the following format:

```
domain action=create name=domain_name
    [comment=comment]
    [{area=area_name}|addAll]
    [superDomain=super_domain_name]
    [InheritPerms[={y|n}]]
```

name

Specifies the name of a CA ITCM domain group. It will be created.

comment

Specifies a comment.

area

Specifies the name of a CA ITCM domain to the CA ITCM domain group to be created.

The parameter can be coded more than once to add a list of CA ITCM domains.

This parameter should not be coded if "addAll" is coded.

addAll

Add all registered CA ITCM domains to the CA ITCM domain group to be created.

This parameter should not be coded if the parameter "area" is coded.

superDomain

Specifies the name of an existing CA ITCM domain group that might contain other CA ITCM domain groups . The CA ITCM domain group to be created will become a subgroup of this CA ITCM domain super group.

If this parameter is not coded, the CA ITCM domain group is placed at the folder named "Domains".

inheritPerms

This parameter specifies if the group being created inherits security permissions or not.
If “InheritPerms” or “InheritPerms=y” is coded then the specified group will inherit permissions to it members and becomes a security group.
If “InheritPerms=n” is coded then the specified group will not inherit permissions to its members and therefore is no security group.
Default: InheritPerms=y .

Note: The action create allows you to create empty CA ITCM domain groups, in other words, neither "area" nor "addAll" have to be coded with the command. CA ITCM domains can be assigned later with the "add (see definition on page 95)" action.

Example:

Assume the domain "dom c" already existing and a new domain "dom e" to be created as a subdomain of "dom c". Furthermore, the areas "area 1" and "area 2" should be added to the domain "dom e". The following command will provide the domain "dom e":

```
cadsmcmd domain action=create name="dom e" area="area 1" area="area 2"
superDomain="dom c"
```

delete—Delete Domain Group

This action deletes a CA ITCM domain group.

This action has the following format:

```
domain  action=delete
    name=domain_name
```

name

Specifies the name of a CA ITCM domain group to be deleted.

Example:

To delete the domain "dom z," the following command should be entered:

```
cadsmcmd domain action=delete name="dom z"
```

list—List Domain (Sub)groups

This action provides a list of CA ITCM domain (sub)groups.

This action has the following format:

```
domain action=list [name=domain_name] [recursive]
```

name

Specifies the name of a CA ITCM domain group. For this group all CA ITCM domains will be listed.

If not coded then all the CA ITCM domain groups of the top level folder "Domains" are listed.

recursive

The CA ITCM domain subgroups contained in the CA ITCM domain group of parameter "name" are listed recursively.

Example:

To list all the domains the following command should be entered:

```
cadsmcmd domain action=list recursive


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
List of domains
--------------------------------------------------------------------------------
dom a (, security group)
..dom b (, security group)
....dom c (, security group)
......dom d (, security group)
......dom e (, security group)
dom d (, security group)
dom f (I'm a comment, security group)
dom g (and i2, security group)

SDCMD<A000000>: OK
```

Example:

To list all the domains at the top level in other words, folder "Domains," the following command should be entered:

```
cadsmcmd domain action=list


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "<default manager>" as user "<default user>" ...OK.

--------------------------------------------------------------------------------
List of domains
--------------------------------------------------------------------------------
dom a  (, security group)
dom d  (, security group)
dom f (I'm a comment, security group)
dom g (and i2, security group)

SDCMD<A000000>: OK
```

Example:

To list all the subdomains of the domain "dom b" the following command should be entered:

```
cadsmcmd domain action=list name="dom b"


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "<default manager>" as user "<default user>" ...OK.


--------------------------------------------------------------------------------
List of domains ("dom b" )
--------------------------------------------------------------------------------
dom c  (, security group)

SDCMD<A000000>: OK
```

Example:

To recursively list all the subdomains of the domain " dom b" the following command should be entered:

```
cadsmcmd domain action=list name=" dom b" recursive


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager " <default manager>" as user " <default user>" ...OK.

------------------------------------------------------------------------------
List of domains (" dom b" )
------------------------------------------------------------------------------
dom c  (, security group)
..dom d (, security group)
..dom e (, security group)

SDCMD<A000000>: OK
```

listMembership—List Domain Supergroups

This action provides a list of CA ITCM domain supergroups that contain the specified one as a CA ITCM domain subgroup.

This action has the following format:

```
domain action=listmembership name=domain_name
```

name

Specifies the name of a CA ITCM domain group which must exist. All CA ITCM supergroups containing this group as a subgroup are listed.

Example:

To list all those domains that contain the domain " dom c" the following command should be entered.

```
cadsmcmd domain action=listMembership name=" dom c"


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off
```

```
Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
"dom c" is subdomain of the following domains
--------------------------------------------------------------------------------
dom b

SDCMD<A000000>: OK
```

modify—Modify Domain Group Attributes

This action modifies the attributes of a CA ITCM domain group.

This action has the following format:

```
domain action=modify name=domain_name
    [newName=new_domain_name]
    [InheritPerms[={y|n}]]
    [comment=comment]
```

inheritPerms

This optional parameter specifies if the group being modified is a security group or not.
If “InheritPerms=n” is coded then the specified group will not inherit permissions to its members and therefore is no security group any longer.
If “InheritPerms” or “InheritPerms=y” is coded then the specified group will inherit permissions to it members and therefore becomes a security group.
If the parameter is not coded then the permission inheritance settings of the group are not changed.

name

Specifies the name of a CA ITCM domain group to be modified.

newName

Specifies the new name of the CA ITCM domain group.

comment

Specifies a comment.

Note: To erase an existing comment, enter the empty string ("").

Example:

Assume that the domain " dom g" should be modified in the following way:

- the new name of the domain is "dom z"
- the comment is set to "I've been modified"

The following command will provide these changes:

```
cadsmcmd domain action=modify name="dom g" newName="dom z" comment="I've been
modified"
```

remove—Remove Domains from Domain Groups

This action removes one or more domains from a specified CA ITCM domain group.

This action has the following format:

```
domain action=remove name=domain_name
    {area=area_name}
```

name

Specifies the name of a CA ITCM domain group from which the CA ITCM domains specified by the "area" parameter will be removed.

area

Specifies the name of a CA ITCM domainthat will be removed from the specified CA ITCM domain group.

The parameter can be coded more than once to remove a list of CA ITCM domains from the group.

Example:

To remove the areas "area 3" and "area 4" from the domain "dom f," the following command should be entered:

```
cadsmcmd domain action=remove name="dom f" area="area 3" area="area 4"
```

removeSubdomains—Remove Domain Groups from Domain Groups

This action removes CA ITCM domain groups as CA ITCM domain groups from the specified CA ITCM domain group.

This action has the following format:

```
domain action=removeSubdomains [name=domain_name}]
    {domain=domain_name}
```

name

Specifies the name of a CA ITCM domain group which must exist and from which the CA ITCM domain subgroups will be removed.

If not coded the CA ITCM domain subgroups are removed from the top level folder "Domains".

domain

Specifies the name of a CA ITCM domain group which must exist and which will be removed as a CA ITCM domain subgroup from the CA ITCM domain group specified in the "name" parameter.

The parameter can be coded more than once to remove a list of CA ITCM domain subgroups.

Note: A remove does not necessarily mean that the CA ITCM domain subgroup to be removed will be deleted. It will only be removed from a list of CA ITCM domain subgroups associated to the specified CA ITCM domain supergroup.

If a CA ITCM domain subgroup "A" is removed from a CA ITCM domain group "B" and if "B" is the only supergroup containing "A", then the remove will fail. "B" can only be deleted.

Example 1:

Assume the domains "dom f" and "dom g" are to be removed as subdomains from domain "dom e". The following command will unlink these domains from "dom e":

```
cadsmcmd domain action=removeSubdomains name="dom e" domain="dom f" domain="dom g"
```

Example 2:

Assume that the domain "dom e" is to be removed from the top level in other words, the folder "Domains". The following command will unlink the domain from this level:

```
cadsmcmd domain action=removeSubdomains domain="dom e"
```

fetch—Fetching Items

The fetch command is valid on enterprise managers only.

Use the fetch command to fetch files from a CA ITCM domain manager or a group of CA ITCM domain managers and store it into the fetch area of the enterprise manager.

This action has the following format:

```
fetch  item=item  [comment=comment]
    path=path_name
    {{area=area_name} | {domain=domain_name} | toAllAreas}
    [cname=name]
    [sendTime="YYYY-MM-DD hh:mm"]
    [haltTime="YYYY-MM-DD hh:mm"]
```

area

Specifies the name of a CA ITCM domain from which the items are fetched.

You can specify this parameter more than once to address a list of CA ITCM domains.

cname

Specifies the the optional unique name of a job container.

If a job container / distribution with the specified name already exists an error will be reported.
If "cname" is not coded, a generic name is generated.

comment

Specifies a comment providing a description of the fetched item or items.

domain

Specifies the name of one or more CA ITCM domain groups from which the items are fetched

You can specify this parameter more than once to address a list of CA ITCM domain groups.

halttime

Specifies the date and time at which a distribution order should be halted. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm".

item

Specifies the name of the items to fetch from the targeted CA ITCM domain managers.

path

Specifies the path to the items to be fetched.

The specifies path name is that on the targeted CA ITCM domain managers.

It has to be coded according to the conventions of the operating system.

To retrieve data from a CA ITCM domain manager, the fetch command uses the parameter "path" to specify what is to be retrieved from the domain manager.

For ease of use the path may contain environment variables defined by the system or user or the configuration files of SD.

(For details, see the *DSM Explorer Help*).

In SD, these environment variables have to enclosed by percent signs "%".

sendTime

Specifies the time to send the associated distribution to the addressed domain managers. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm". If the parameter is not coded, the actual date and time is used.

toAllAreas

If coded the related distribution will be sent to all CA ITCM domains registered at the enterprise manager.

Use of the % Sign

Assume that on both, the enterprise and the domain manager, the environment variable %SAMPLES% is defined, but on the enterprise the variable is pointing to "c:\samples", while on the domain manager it is pointing to "g:\examples".

All txt files should be fetched. Now launching the following command from the Windows command prompt

```
cadsmcmd fetch item=sample path=%SAMPLE%\*.txt area=domain_manager
```

will result in an error or unexpected results. Because SAMPLES is defined on the enterprise manager the command prompt will resolve the variable and launch the cadsmcmd with the following path parameter:

```
path=c:\samples\*.txt
```

To avoid the variables resolution by the Windows command prompt the percent sign has to be escaped from being interpreted by the command prompt. This is done by preceding the percent sign with a caret "^".

Therefore, in the following invocation, all parameters will by passed unmodified by the command prompt to the cadsmcmd and the environment variable SAMPLES will be resolved at the domain manager's site:

```
cadsmcmd fetch item=samples path=^%SAMPLES^%\*.txt area=domain_manager
```

Example

The following example illustrates how the "fetch" command can be used to fetch "Software TestC" from a Windows domain manager.

```
cadsmcmd fetch item="Software TestC" comment="Fetches Software TestC"
path=c:\test\swtest3 area=N_LOCAL_SRV cname=TestCont8
sendtime="2006-03-07 22:15" halttime="2006-03-08 03:30"
```

The following fetch example is similar to the previous example; however, environment variables are used to indicate the path.

```
cadsmcmd fetch item="Software TestC" comment="Fetches Software TestC"
path=%FETCHSW%\*.exe area=N_LOCAL_SRV cname=FetchContainer
sendtime="2006-03-07 22:15" halttime="2006-03-08 03:30"
```

halt—Pausing an Active Distribution

The halt command is valid only when issued from the enterprise manager command line.

Halt is used to halt an active distribution. The distribution can be subsequently deleted or renewed.

This command has the following format:

```
halt name=job_container_name
```

name

Specifies the name of the job container to halt.

Example:

Consider the following halt example:

```
cadsmcmd halt name=Testcont11
```

To view the progress of the halt request, issue the following progress command:

```
cadsmcmd progress name=TestCont11
```

which produces a status similar to the following:

```
Distribution name: TestCont11, Status Halt in Progress.
SendDateTime: 2003/03/05 17:41, InstallDateTime: 2003/03/05 17:41,
HaltDateTime: 2003/03/05 17:42
Nr of Areas Total: 2, Waiting: 0, Active; 1, in Error: 0, OK:1.

Area: N_LOCAL_SRV at Downtown 100 % sent. Status Active.
Area: N_LOCAL_SRV_02 at Rural Area 0 % sent. Status Active.
SDCMD<A000000> : OK
```

image—Manage OS Images

This section contains the following topics:

assignToTargets—Assign OS Images to Targets

This action assigns the specified image to a list of targets and creates new planned configurations with this image for the targets or modifies the possibly already existing ones.

The syntax of the action is as follows:

```
image   action=assignToTargets
    name=image_name
    {computer={OSIM_target_name} |
    compgrp={computer_group_name} |
ccompgrp={computer_group_name sep scope}}
```

name

Specifies the name of an image to be installed at the specified targets

computer

Specifies the name of the OSIM targets at which the specified image can be installed. The option can be specified more than once to address a list of targets. The option should not be coded together with the option “compgrp” or “ccompgrp”.

compgrp

Specifies the name of a computer group its members that are OSIM managed systems are taken as targets for the specified image. The option can be specified more than once to address a list of groups. The option should not be coded with the option “computer”.

ccompgrp

Specifies the computer group to be addressed as a pair of group name and scope. By default, name and scope are separated by a dot ("."), but an alternate separator can be specified with the sep option. The separator should not be used as part of the names.

The option can be specified more than once to address a list of groups.

The option should not be coded with the option “computer”.

scope

For the group scope only the following values are valid:

local

The group is created at the local domain manager,

global

The group has been created at the enterprise manager and may be replicated

sep

Specifies an alternate separator used with the ccompgrp option. The separator should not be part of the group name coded with the ccompgrp options

Note: Only registered images can be assigned to targets. If an image is detected, only the error BMSAPI_IMAGE_DETECTED – “Image not registered but detected only” is reported for this image.

delete—Delete Images from the OSIM Database

This action deletes the specified images from the OSIM database and the possible related SD packages too.

The syntax of the action is as follows:

```
image    action=delete
    {[images=image_identifier,...]}
    [sep=separator_sign]
```

images

Specifies the name of an image to be deleted from the OSIM database. With the removal of the image from OSIM also a possible SD registration of the related package is deleted too. The option can be coded more than once to specify a list.

The image identifier has the following structure:

```
{«image name» | («image name»«separator sign»{bootImage | osImage})}
```

If neither bootImage nor osImage is coded then osImage is assumed. The separator sign is a dot by default but can be redefined by the sep argument. The separator sign should not appear in the image names.

sep

Specifies an alternate separator used with the image identifiers. The separator should not be part of the image names.

Note: You should be aware of the following:

- An OS image is not deleted if it is assigned to any OSIM configuration. The error BMSAPI_IMAGE_INUSE – “Image in use” is reported. Use action listConfigs for determining the configurations in that it is used.

  Boot images are deleted in any way.
- The deletion process does not check whether an image is deployed at a boot server or not. When it is deployed then it is reported as detected later on.
- For deleting an image the current user owning the session needs delete permissions for this image.

list—List Images Registered at OSIM Manager

This action list provides a list of boot and OS images currently registered at the OSIM manager.

The syntax of the action is as follows:

```
image    action=list
    [name=image_name]
    [{bootImage | osImage}]
    [detailed]
```

name

Specifies a name of an image. Only the images that meet the specified name are listed.

If not coded no name check takes place.

{bootImage | osImage}

if bootImage is coded then the images reported have to be boot images, if only osImage is coded then the images reported have to be OS images, and if none of them is coded then both types of images are reported.

detailed

If this option is coded then also the default settings for the boot parameters are listed.

The command provides an output list on stdout of the following format:

```
-------------------------------------------------------------------
[Detailed] List of Images [({bootImage|osImage})]
[Name: <name>]
-------------------------------------------------------------------
Image name...................: «name of image 1»
...Image type................: {boot image | OS image («osType»)}
...Status....................: (registered | detected}
...Image FIPS capability.....: (compliant  | noncompliant)
...SD package name...........: «sd package name of the image»
...SD package version........: «sd package version of the image»
...SD package comment........: «comment stored with the package»
[...Locale...................: «the locale of the image»
...Batch file................: «name of the batch file»]
...Support Boot Image Type....: <<Support boot image type and boot server mode>>
[...Parameters
......<name of param 1>.......: «default setting for parameter 1»
.........................
.........................
.........................]
......<name of para m>........: «default setting for parameter m»
.........................
.........................
.........................]
Image name...................: «name of image n»
.........................
.........................
.........................
```

The information in parenthesis is available for OS images only. Further more the information about parameters is only printed when "detailed" is coded. If the status is "detected" all subsequent information is skipped.

listDeployed—List Boot Servers where OSIM Image Is Deployed

This action provides a list of those boot servers where the specified boot or OS image is deployed.

The syntax of the action is as follows:

```
image   action=listDeployed
    name=name_of_an_image
    [{bootImage | osImage}]
```

name

Specifies the name of an image for which the list of all boot servers is listed where it is deployed.

{bootImage | osImage}

These options specify whether the image in question is a boot or an OS image. If "bootImage" is coded then a boot image is assumed. Otherwise an OS image is assumed.

The provided list has the following format:

```
------------------------------------------------------------
List of boot servers where («name of image»,{boot|os}image) is deployed
------------------------------------------------------------
Image name.....................................: «name of image»
..Image type...................................: {boot image | OS image}
..Status.......................................: {registered | detected}
..Boot server name.............................: «name of boot server 1»
.........
.........
.........
.............................................: «name of boot server m»
.........
.........
.........
```

listConfigs—List Targets and Configurations of OS Image

This action provides a list of those targets and their configurations associated with the specified image.

The syntax of the action is as follows:

```
image   action=listConfigs
    name=image_name
```

name

This option specifies the name of an OS image for that all the targets and configurations are listed the image is associated with.

The provided list is of the following format:

```
------------------------------------------------------------
List of Configurations
------------------------------------------------------------
Image name.............................: «name of OS image 1»
..Target name..........................: «name of OSIM target 1»
....Configuration Type..................: {planned | scheduled |
.........................................current}
[....Status ............................: {activated|analysing|
.........................................pending|installing|
.........................................error|cancelpending|
.........................................stopped}]
....•
....•
....Configuration Type..................: {planned | scheduled |
.........................................current}
[...Status..............................: {activated|analysing|
.........................................pending|installing|
.........................................error|cancelpending|
.........................................stopped}]
..•
..•
..•
..Target name..........................:«name of OSIM target n»
....Configuration Type..................: {planned | scheduled |
.........................................current}
..•
..•
..•
```

The status block and the due date is only printed when the configuration type is “scheduled”.

modify—Modify OSIM Image

This action modifies the default settings of a parameter for the specified image.

The syntax of the action is as follows:

```
image    action=modify
    name=image_name
    paramName=parameter_name
    paramValue=new_default_setting
```

name

Specifies the name of the image of that the default for a parameter will be changed.

paramName

Specifies the name of the parameter of that the default is changed.

The defaults for the following parameters can not be changed:

- BatchFile
- HostName
- MACAddress
- OSImage

paramValue

Specifies the new default.

Note: You should be aware of the following:

- A default setting can only be changed as long as the image is not assigned to any target system.
- For changing the default of an image the current user owning the session needs change permissions on the image.
- For removing settings just enter the empty string.

deploy—Deploy OSIM Images to Boot Servers

This action deploys the images specified to the boot servers specified. The cadsmcmd creates SD install orders to deploy the images at the boot server. The job container used and the install orders created and their mapping on the images are reported on stdout.

The syntax of the action is as follows:

```
image   action=deploy
    {images=image_identifier, ...}
    [sep=separator_sign]
    [procedure=boot_server_staging_procedure]
    {bootServers=boot_server_name, ...}
    [cname=name_of_job_container_used]
    [{noLinkage | transaction [rollback] [noCascade]| synchronized [noCascade]}]
    [jcPriotrity=job_container_priority]
    [unsealed]
    [atTime=YYYY-MM-HH hh:mm]
    [deliveryTime=YYYY-MM-HH hh:mm]
    [calendarName=name_of_the_delivery_calendar]
```

images

Specifies the name of an image to be deployed at the specified boot servers. The option can be coded more then once for specifying a list of images.

The image identifier has the following structure:

```
{«image name» |
(«image name»«separator sign»{bootImage | osImage})}
```

If neither "bootImage" nor "osImage" is coded then "osImage" is assumed.

The separator sign is a dot by default but can be redefined by the "sep" argument. The separator sign should not appear in the image names.

sep

Specifies an alternate separator used with the image identifiers. The separator should not be part of the image names.

procedure

This parameter specifies the procedure used by SD to install the image at the specified boot servers. The default procedure is “Add to BootServer” for OS images and “Create Boot Image” for boot images.

bootServers

Specifies a boot server the image list is deployed to. The option can be coded more than once to specify a list of boot servers.

cname

This option specifies the name of the job container used for staging the required images. If the container does not exist then it is created according to the attributes specified. If it already exists then it has to be unsealed, otherwise the action results in an error. The attributes of an existing container are not modified.

noLinkage

The result of a job does not have any impact on the other jobs execution.

(default)

transaction

The jobs of the container are processed in sequence. Once a job fails for a target then no subsequent jobs of the container are processed for this target.

synchronized

The jobs of the container are processed in sequence. Once a job fails for any target then no subsequent jobs of the container are processed for any target.

rollback

Once a job of the container fails a rollback is performed for all processed jobs of the container for the affected target as far as they allow it.

noCascade

The dependencies of the jobs are not resolved during container evaluation.

jcPriotrity

Specifies the priority of the job container. The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded priority 5 is assumed.

unsealed

The job container remains unsealed after the generated jobs have successfully been created.

atTime

Specifies the date and time when the action will be started. The date has the ISO format "YYYY-MM-DD hh:mm". If not coded the current date/time is taken.

deliveryTime

Specifies the start time of delivery from the domain manager.

The date has the ISO format "YYYY-MM-DD hh:mm". If not coded the current date/time is taken.

calendarName

Specifies the name of the calendar that controls the time when the evaluation can take place.

Output

The command returns on stdout:

```
The commandFollowing job container used for staging: “«job container name»”
«image name 1» - “«item 1»;«version 1»;«procedure 1»”: «return code»
.....................
.....................
.....................
«image name n» - “«item n»;«version n»;«procedure n»”: «return code»
```

Note: To do so it is necessary that the image is registered in SD as well as in OSIM and that the OSIM registration refers correctly to the SD registration. If the reference is not found or invalid the function records the problem with following message:

```
SDCMD<A000101> Software does not exist (<<product name>> - <<product version>> -
<procedure> - <procedure type>)
```

remove—remove OSIM Images from Boot Servers

This action removes the images specified from the boot servers specified. The cadsmcmd creates SD uninstall orders to clear the images at the boot server. The job container used and the uninstall orders created and their mapping on the images are reported on stdout and allow to monitor the clearing progress with SD methods.

The syntax of the action is as follows:

```
image   action=remove
    {images=image_identifier, ...}
    [sep=separator_sign]
     [procedure=boot_server_staging_procedure]
    {bootServers=boot_server_name, ...}
    [cname=name_of_job_container_used]
    [{noLinkage | transaction [rollback] [noCascade]| synchronized [noCascade]}]
    [jcPriotrity=job_container_priority]
    [unsealed]
    [atTime=YYYY-MM-HH hh:mm]
    [deliveryTime=YYYY-MM-HH hh:mm]
    [calendarName=name_of_the_delivery_calendar]
```

images

Specifies the name of an image to be removed from the specified boot servers. The option can be coded more then once for specifying a list of images.

The image identifier has the following structure:

```
{«image name» |
(«image name»«separator sign»{bootImage | osImage})}
```

If neither "bootImage" nor "osImage" is coded then "osImage" is assumed. The separator sign is a dot by default but can be redefined by the sep argument. The separator sign should not appear in the image names.

sep

Specifies an alternate separator used with the image identifiers. The separator should not be part of the image names.

procedure

This parameter specifies the procedure used by SD to remove the image from the specified boot servers. The default procedure is “Remove from BootServer” for OS images and “Remove Boot Image” for boot images. These default names are not localizable.

bootServers

Specifies a boot server the image list is removed from. The option can be specified more than once for allowing lists of boot servers.

cname

This option specifies the name of the job container used for staging the required images. If the container does not exist then it is created according to the attributes specified. If it already exists then it has to be unsealed, otherwise the action results in an error. The attributes of an existing container are not modified.

noLinkage

The result of a job does not have any impact on the other jobs execution.

(default)

transaction

The jobs of the container are processed in sequence. Once a job fails for a target then no subsequent jobs of the container are processed for this target.

synchronized

The jobs of the container are processed in sequence. Once a job fails for any target then no subsequent jobs of the container are processed for any target.

rollback

Once a job of the container fails a rollback is performed for all processed jobs of the container for the affected target as far as they allow it.

noCascade

The dependencies of the jobs are not resolved during container evaluation.

jcPriotrity

Specifies the priority of the job container. The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded priority 5 is assumed.

unsealed

The job container remains unsealed after the generated jobs have successfully been created.

atTime

Specifies the date and time when the action will be started. The date has the ISO format "YYYY-MM-DD hh:mm". If not coded the current date/time is taken.

deliveryTime

Specifies the start time of delivery from the domain manager.

The date has the ISO format "YYYY-MM-DD hh:mm". If not coded the current date/time is taken.

calendarName

Specifies the name of the calendar that controls the time when the evaluation can take place.

Output

The command returns on stdout:

```
Following job container used for removing: “«job container name»”
«image name 1» - “«item 1»;«version 1»;«procedure 1»”: «return code»
.....................
.....................
.....................
«image name n» - “«item n»;«version n»;«procedure n»”: «return code»
```

Note: To do so it is necessary that the image is registered in SD as well as in OSIM and that the OSIM registration refers correctly to the SD registration. If the reference is not found or invalid the function records the problem with following message:

```
SDCMD<A000101> Software does not exist (<<product name>> - <<product version>> -
<procedure> - <procedure type>)
```

install - configure - activate - uninstall—Generate and Execute Jobs in Job Containers / Distributions

These commands are used on the enterprise or the domain manager to generate an *install*, *configure*, *activate*, or *uninstall* job in a job container and execute it.

On the enterprise manager, this job container is assigned to a distribution.

This command has the following format on the domain manager:

```
{install|configure|activate|uninstall}
    item=item_name
    version=item_version
    procedure=procedure_name
    [installedWith=install_procedure]
    {{computer=computer_name}
    |{{compgrp=computer_group_name
    | ccompgrp=(computer_group_name.domain_name)}}}
    [sep=separator_sign]
    [cname=name]
    {nolinkage|transaction|synchronized}
    [rollback=y|n]
    [cascade=y|n]
    [unsealed]
    after={exacttime|boottime}
    [globalTime=y|n]
    [deliveryTime="YYYY-MM-DD hh:mm"]
    [atTime="YYYY-MM-DD hh:mm"]
    [promptUser[={y|n}]]
    [allowCancel[={y|n}]]
    [execTimedOut[={y|n}]]
    [prompt=days.hours]
    [{preaction={none|reboot|logoff}|bootbefore}]
    [{postaction={none|reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}|boota
    fter}]
    [{jobTimeOut=days.hours | timeout=hours}]
    [offline[={y|n}]]
    [runAtShutdown[={y|n}]]
    [preventLogon[={y|n}]]
    [resolveQuery[={y|n}]]
    [calendarname=delivery_calendar]
    [nocalendar[={y|n}]]
    [stagingServer[={y|n}]]
    [parameters=user_parameters]
    jcPriority={1-10}
    [reinstall[={y|n}]]
    [userJobMessage]
    [userMessage]
```

This command has the following format on the enterprise manager:

```
{install|configure|activate|uninstall}
    item=item_name
    version=item_version
    procedure=procedure_name
    [installedWith=install_procedure]
    |{{compgrp=computer_group_name
    | ccompgrp=(computer_group_name.domain_name)}}}
    [sep=separator_sign]
    after={exacttime|boottime}
    [globalTime=y|n]
    {{area=area_name} | {domain=domain_name} | toAllAreas}
    [deliveryTime="YYYY-MM-DD hh:mm"]
    [atTime="YYYY-MM-DD hh:mm"]
    cname=[name]
    {nolinkage|transaction|synchronized}
    [rollback=y|n]
    [cascade=y|n]
    [sendTime="YYYY-MM-DD hh:mm"]
    [haltTime="YYYY-MM-DD hh:mm"]
    [promptUser[={y|n}]]
    [allowCancel[={y|n}]]
    [execTimedOut[={y|n}]]
    [prompt=days.hours]
    [{preaction={none|reboot|logoff}|bootbefore}]
    [{postaction={none|reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}|boota
    fter}]
    [{jobTimeOut=days.hours | timeout=hours}]
    [offline[={y|n}]]
    [runAtShutdown[={y|n}]]
    [preventLogon[={y|n}]]
    [resolveQuery[={y|n}]]
    [calendarname=delivery_calendar]
    [nocalendar[={y|n}]]
    [stagingServer[={y|n}]]
    [parameters=user_parameters]
    [jcPriority={1,...,10}]
    [distPriority={1,...,10}]
    [reinstall[={y|n}]]
    [userJobMessage]
    [userMessage]
```

activate

Activate job for the specified item. The associated task of the specified procedure has to be of the type "activate"

after

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

allowCancel

If "allowCancel" or "allowCancel=y" is coded, the user will be granted to cancel the job's execution.
If "allowCancel=n" is coded, the user will not be granted.

If the parameter is not coded, the default is given by the related procedure's job option value.

area

Specifies the name of a CA ITCM domain as a target for distribution.

You can specify this parameter more than once to address a list of CA ITCM domains.

(enterprise manager only).

atTime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

calendarname

Specifies the name of the calendar that controls the time when the evaluation can take place.

cascade

If "cascade" or "cascade=y" is coded, the job linkage of the container is switched to resolution of dependencies into install cascades.
If "cascade=n" is coded, the job linkage is switched to no resolution.

cname

Specifies the the optional unique name of a job container.

If a job container / distribution with the specified name already exists an error will be reported.
If "cname" is not coded, a generic name is generated.

Notes:
On domain manager:
If a job container of the specified name already exists and is unsealed, then the generated job will be assigned to this container. If the container is already sealed an error is indicated.
If a job container of the specified name does not yet exist, it will be created according to the job linkage options specified.
If cname is not coded, a generic name is generated.

On enterprise manager:
"cname" is used to name the distribution and the related job container. Both are expected to be nonexistent and will be created. The job container is created according to the job linkage options specified.
If "cname" is not coded a generic name is generated.

compgrp

Specifies the name of a computer group. It will be addressed by the job.

The parameter can be coded more than once to assign a list of computer groups.

ccompgroup

Specifies the computer group to be addressed as a pair of group name and scope. By default, name and scope are separated by a dot ("."), but an alternate separator can be specified with the "sep" option. The separator should **not** be used as part of the names.

For the group scope only the following values are valid:

local

The group is created at the local domain manager,

global

The group has been created at the enterprise manager and may be replicated.

The "ccompgrp" can be coded with the "compgrp" in one call.

computer

Specifies the name of the computer. It will be addressed by the job.

The parameter can be coded more than once to assign a list of computers.

(domain manager only)

configure

Generates a configuration job for the specified item. The associated task of the specified procedure has to be of the type "configure".

deliveryTime

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

distPriority

Specifies the priority that will be assigned to the distribution.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded priority 5 is assumed.

domain

Specifies the name of a CA ITCM domain group to which the distribution will be sent for execution.

You can specify this parameter more than once to address a list of CA ITCM domain groups.

(enterprise manager only)

execTimedOut

If "execTimedOut " or "execTimedOut=y" is coded, the job's execution will be automatically started when the user prompt times out.
If "execTimedOut=n", the job will not automatically be started.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

haltTime

Specifies the date and time at which a distribution order should be halted. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm".

install

Generates an installation job for the specified item. The associated task of the specified procedure has to be of the type "install".

installedWith

Specifies the name of the install procedure that has been used to install the specified item at the target systems.

This parameter has to be coded for the commands "activate", "configure", and "uninstall". For the command "install" it is meaningless and will be ignored.

item

Specifies the name of the item the job will be generated for.

jcPriority

Specifies the priority of the job container.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded priority 5 is assumed.

jobTimeOut=days.hours

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

noCalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

noCascade

The job linkage option is set to "Ignore cascading".

This parameter is allowed only if "transaction" or "synchronized" are coded.

noLinkage

Run the job independently of the other jobs in the container.

offline

If "offline" or "offline=y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.
If "offline=n" the connection will not be released.

parameters

Specifies the user parameters for the procedure.
Multiple user parameters must be separated by spaces inside double quotes.
If an empty string is coded, then no parameters are assumed.

postaction

Specifies the necessary actions that must take place after completion of the specified procedure. The following values are valid:

none

Performs no post-action.

reboot

Restarts the system after completion of the procedure.

logoff

Logs off the user after completion of the procedure.

rebootAtEnd

Restarts the system after completion of all the jobs of this container.

logoffAtEnd

Logs off the user after completion of all the jobs of this container.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

preaction

Specifies that necessary actions must take place before the specified procedure starts. The following values are valid:

none

Performs no pre-action.

reboot

Restarts the system before the start of the procedure.

logoff

Logs off the user before the start of the procedure.

preventLogon

If "preventLogon" or "preventLogon=y" is coded, then the user logon will be rejected during the job's runtime. If a user is already logged on, the execution of the job is delayed until the user logs off.
If "preventLogon=n" the user logon will not be rejected.

procedure

Specifies the name of a procedure to be run.

prompt=days.hours

Specifies the time period in which the user will be prompted for the job's start.

The period's format is "*d.h*" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0). If the specified value falls below the valid minimum, the value is replaced by the minimum. If the specified value exceeds the valid maximum, then the value is replaced by the maximum. No warning is given.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

reinstall

Optional parameter. Default: “reinstall=n”

If “reinstall” or “reinstall=y” is coded then a possible installation record related to the order is removed before the job is evaluated by the domain manager.

If “reinstall=n” is coded then such records will remain.

Note: The reinstall option is valid for the install action only but not for the configure, activate, or unistall actions.

resolveQuery

Evaluate a possibly related query group before the job is generated.

rollback

Set the job linkage option to "Enable Transaction".

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

sendTime

Specifies the time to send the associated distribution to the addressed domain managers. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm". If the parameter is not coded, the actual date and time is used.

sep

Specifies an alternate separator used with the "ccompgroup" option. The separator should not be part of the group name code with the "ccompgroup" options.

stagingServer

Deliver the related product also to the staging libraries of the scalability servers of the target computers.

This parameter applies to procedures of task type "install" only!

synchronized

The job linkage option is set to "Synchronized job execution".

timeout

Specifies the timeout specifies the expiration period of a job in the form “h” where h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
timeout should not be coded with the jobTimeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 168

toAllAreas

If coded the related distribution will be sent to all CA ITCM domains registered at the enterprise manager.

transaction

The job linkage option is set to "Batch job execution".

uninstall

Generates an uninstall job for the specified item. The associated task of the specified procedure has to be of the type "uninstall".

unsealed

The addressed job container remains unsealed after the generated job has successfully been created.

version

Specifies the version of the item the job will be generated for.

userJobMessage

Specifies the custom administrator message for a job in a job container.

userMessage

Specifies the custom administrator message for a job container.

The parameters "allowCancel," "execTimedOut" and "prompt" are valid, if the "promptUser" option is already set or is set by this command using the "promptUser" parameter.
If neither the "promptUser" option of a job nor the "promptUser" parameter of the command is set, these parameters will be ignored.
No warning is given.

jobcontainer—Job Container Commands

All the methods of this chapter are for domain manager only.

A job container is a unit of work grouping a number of jobs and associating them with common attributes of execution.

This section contains the following topics:

General Job Container Commands

This section contains the following topics:

activate—Activate Job Container

This action activates the specified job container that must already be evaluated.

This action has the following format:

```
jobcontainer action=activate name=jobcontainername
```

Name

Specifies the name of the job container to be activated.

Example:

Suppose the job container "myContainer" is sealed and evaluated. It is now ready to be executed. The following command will schedule the container for execution:

```
cadsmcmd jobContainer action=activate name="myContainer"
```

create—Create Job Container

This command creates an empty job container.

This command has the following format:

```
jobcontainer action=create name=job_container_name
    [{noLinkage|{[transaction][rollback]|synchronized}
    [noCascade]}]
    [jcPriority={1,...,10}]
    [userMessage]
```

jcPriority

Specifies the priority of the job container.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded priority 5 is assumed.

name

Specifies the name of the job container to be created.

noCascade

The job linkage option is set to "Ignore cascading".

This parameter is allowed only if "transaction" or "synchronized" are coded.

noLinkage

Run the job independently of the other jobs in the container.

rollback

Set the job linkage option to "Enable Transaction".

synchronized

The job linkage option is set to "Synchronized job execution".

transaction

The job linkage option is set to "Batch job execution".

userMessage

Specifies the custom administrator message for a job container.

Note: The default characteristic for a jobcontainer is (transaction, norollback, cascade):

- Transaction: a batch job execution where the jobs are sequentially processed and if any job fails no remaining job in the sequence will be processed.
- No rollback: In case of a failure the already performed jobs will not be automatically rolled back.
- Cascade: Dependencies will be resolved to cascades of installation.

Example:

Suppose a job container named "4 sample" has to be created. The container should meet the following requirements:

- The jobs of the container should be executed at the target in sequence. If a job fails at a target no subsequent job of the container should be executed at this target and already completed jobs of this container should be rolled back.
- If there are dependencies for the job, they will not be resolved. There will be no cascades.

The following command will create the job container:

```
cadsmcmd jobContainer action=create name="4 sample" transaction rollback nocascade
```

delete—Delete Job Container

This action deletes the specified job container.

This action has the following format:

```
jobcontainer action=delete name=jobcontainername
```

name

Specifies the name of the job container to be deleted.

Example:

Suppose the job container "myContainer" is obsolete and should be deleted.

The following command will delete the container:

```
cadsmcmd jobContainer action=delete name="myContainer"
```

halt—Halt Job Container

This action suspends the deliveries of the specified job container from execution.

This action has the following format:

```
jobcontainer action=halt name=jobcontainername
```

name

Specifies the name of the job container to be halted.

Note: A delivery is an order to store a product at the software library of a scalability server.

Example:

Suppose the activated job container "myContainer" contains delivery orders for scalability servers, but a problem has arisen that makes it necessary to suspend the delivery.

The following command suspends the deliveries from execution:

```
cadsmcmd jobContainer action=halt name="myContainer"
```

list—List Job Container

This action lists all the required jobcontainers. The list supports the feature of "renewed states". The first status information of a job container is its primary state information, while the second one is from the "renewed state".

This action has the following format:

```
jobcontainer action=list
    [{filter=filter|filterfile=file_name}]
```

filter

Specifies the expression to confine the amount of information listed

The following attributes are valid to build atomic filter expressions:

Attribute	Type
Job container name	string
Job container state	{created | waiting | evaluating | cascading | building | in progress | successfully completed | failed | terminating | successfully built | build completed with failures | warning | halt in progress | halted}
Job container renew state	{created | waiting | evaluating | cascading | building | in progress | successfully completed | failed | terminating | successfully built | build completed with failures | warning | halt in progress | halted}

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command.

Example:

```
Cadsmcmd jobcontainer action=list
```

The following information is shown:

```
--------------------------------------------------------------------------------
List of job containers
--------------------------------------------------------------------------------
HHHHH01B [2/4/2003 03:50:26 PM] (successfully completed)
SD scalability servers [04.02.2003 15:42:39] (failed) (in progress)
--------------------------------------------------------------------------------
Number of job containers found: 2
Number of job container shown:  2
```

To list the job containers starting with "HH", enter the following command:

```
cadsmcmd jobcontainer action=list filter="Job container name=HH*"
```

The following information is shown.

```
--------------------------------------------------------------------------------
List of job containers
--------------------------------------------------------------------------------
HHHHH01B [2/4/2003 03:50:26 PM] (successfully completed)
--------------------------------------------------------------------------------
Number of job containers found: 2
Number of job containers shown: 1
SDCMD<A000000>: OK
```

listFailed—List Failed Job Containers

This action lists all the specified job containers of state failed.

The first status information shown for a job container is its primary state information while the second one is from the "renewed state".

This action has the following format:

```
jobcontainer action=listfailed
    name=job_container_name
    [{filter=filter|filterfile=file_name}]
```

name

Specifies the name of the job container to be evaluated.

filter

Specifies the filter to confine the amount of job containers listed.

The attributes for building filter atomic expressions is listed under the section list (see page 134).

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command..

Example:

To receive the list of failed containers, enter the following command:

```
cadsmcmd jobContainer action=listFailed
```

The following information is shown:

```
--------------------------------------------------------------------------------
List of job containers
--------------------------------------------------------------------------------
jc_DeliveryTrace (failed)
jc_DeliveryTrace_SS (failed)
jc_ss_wwwww (failed)
--------------------------------------------------------------------------------
Number of job containers found:........4
Number of job container shown:........3

SDCMD<A000000>: OK
```

To receive the list of failed containers, starting with jc_d* using filters, enter the command:

```
cadsmcmd jobcontainer action=listfailed filter="Job container name=jc_D*"
```

The following information is shown:

```
--------------------------------------------------------------------------------
List of job containers
--------------------------------------------------------------------------------
jc_DeliveryTrace (failed)
jc_DeliveryTrace_SS (failed)
--------------------------------------------------------------------------------
Number of job containers found:........4
Number of job container shown:........2
SDCMD<A000000>: OK
```

listrenewals—List Renewals

This action allows you to list the renewals related with a job container.

This action has the following format:

```
jobContainer action=listRenewals
    name=job_container_name
    [{filter=filter|filterfile=file_name}]
```

name

Specifies the name of the job container and the renewals to be listed.

filter

Specifies the expression to confine the amount of information listed.

The attributes for building filter atomic expressions is listed under the section list (see page 134).

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command..

Example:

To list all the renewals associated with the job container "myContainer" enter the following command:

```
cadsmcmd jobContainer action=listRenewals name="myContainer"
```

The following information is shown:

```
--------------------------------------------------------------------------------
List of renewals for job container "myContainer"
--------------------------------------------------------------------------------
jcnto1 [12/15/2008 11:34:00 AM] (in progress)
jcnto1 [12/16/2008 10:34:00 AM] (in progress)
--------------------------------------------------------------------------------
Number of job containers found: 2
Number of job containers shown: 2
SDCMD<A000000>: OK
```

To list all the renewals associated with the job container "jcnt01" using filters, enter the following command:

```
cadsmcmd jobcontainer action=listrenewals name=jcnt01 filter="Job container
name=jcnt01[12/15*"
```

The following information is shown:

```
--------------------------------------------------------------------------------
List of renewals for job container "jcnt01"
--------------------------------------------------------------------------------
jcnt01[12/15/2008 11:15:56 AM] (failed)
--------------------------------------------------------------------------------
Number of job containers found: 2
Number of job containers shown: 1
SDCMD<A000000>: OK
```

modfiy—Modify Job Container

This command lets you modify the attributes of an existing unsealed job container.

This command has the following format:

```
jobcontainer action=modify name=job_container_name
    [newName=new_job_container_name]
    [{nolinkage|transaction|synchronized}]
    [rollback[={y|n}]]
    [cascade[={y|n}]}]
    [jcPriority={1,...,10}]
    [userMessage]
```

cascade

If "cascade" or "cascade=y" is coded, the job linkage of the container is switched to resolution of dependencies into install cascades.
If "cascade=n" is coded, the job linkage is switched to no resolution.

jcPriority

Specifies the new priority that will be assigned to the job container.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded the priority is not changed.

name

Specifies the name of the job container to be modified.

newName

Specifies the new name of the job container.

noLinkage

Run the jobs of the container independently of each other.

rollback

Set the job linkage option to "Enable Transaction".

synchronized

The job linkage option is set to "Synchronized job execution".

transaction

The job linkage option is set to "Batch job execution".

userMessage

Modifies the custom administrator message of a job container.

Note: To modify a job container it has to be unsealed.

Attributes remain unchanged when none of their related parameters is coded.

Example:

Suppose the job container "4 sample" should be changed in the way that there are no linkages between the jobs any longer and the name should be changed to "myContainer" . The following command will provide these changes:

```
cadsmcmd jobContainer action=modify name="4 sample" newName="myContainer" nolinkage
```

recover—Recover Failed Jobs

This command uninstalls the failed jobs of a specified container and cleans up the target computers. The job options of the recover jobs are modified on the basis of the related failed jobs (according to the parameters and their coded values). Options for the parameters that are not coded remain unchanged for the uninstall jobs.

This command has the following format:

```
jobcontainer action=recover name=job_container_name
    [newName=new_container_name]
    [{transaction|synchronized|noLinkage}]
    [rollback[={y|n}]]
    [cascade[={y|n}]]
    [deliveryTime="YYYY-MM-DD hh:mm"]
    [atTime="YYYY-MM-DD hh:mm"]
    [after={exacttime|boottime}]
    [calendarname=delivery_calendar
    [preaction={reboot|logoff}]
    [postaction={reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}]
    [promptUser[={y|n}]]
```

```
[allowCancel[={y|n}]]
[execTimedOut[={y|n}]]
[prompt=days.hours]
[offline[={y|n}]]
[runAtShutdown[={y|n}]]
[preventLogon[={y|n}]]
[globalTime[={y|n}]]
[nocalendar[={y|n}]]
[resolveQuery[={y|n}]]
[parameters=user_parameters]
[jobTimeout=d.h]
[userMessage]
```

after

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

allowCancel

If "allowCancel" or "allowCancel=y" is coded, the user will be granted to cancel the job's execution.
If "allowCancel=n" is coded, the user will not be granted.

This parameter should only be coded, if "promptUser" is already enabled or will be enabled with this call, otherwise this setting will be ignored. No warning is given.

atTime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

calendarName

Specifies the name of the calendar that controls the time when the evaluation can take place.

cascade

If "cascade" or "cascade=y" is coded, the job linkage of the container is switched to resolution of dependencies into install cascades.
If "cascade=n" is coded, the job linkage is switched to no resolution.

deliveryTime

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

execTimedOut

If "execTimedOut " or "execTimedOut=y" is coded, the job's execution will be automatically started when the user prompt times out.
If "execTimedOut=n", the job will not automatically be started.

This parameter should only be coded, if "promptUser" is already enabled or will be enabled with this call, otherwise this setting will be ignored. No warning is given.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

jobTimeout

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

name

Specifies the name of the job container to be recovered.

newName

Specifies the new name of the job container which will contain the recover jobs.

If this parameter is not coded a default container name, of the form "Recover[<timestamp>] <name of the container to be recovered>" is generated.

noCalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

noLinkage

Run the job independently of the other jobs in the container.

offline

If "offline" or "offline=y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.
If "offline=n" the connection will not be released.

parameters

Specifies the user parameters for the procedure.
Multiple user parameters must be separated by spaces inside double quotes.
If an empty string is coded, then no parameters are assumed.

postaction

Specifies the necessary actions that must take place after completion of the specified procedure. The following values are valid:

none

Performs no post-action.

reboot

Restarts the system after completion of the procedure.

logoff

Logs off the user after completion of the procedure.

rebootAtEnd

Restarts the system after completion of all the jobs of this container.

logoffAtEnd

Logs off the user after completion of all the jobs of this container.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

preaction

Specifies that necessary actions must take place before the specified procedure starts. The following values are valid:

none

Performs no pre-action.

reboot

Restarts the system before the start of the procedure.

logoff

Logs off the user before the start of the procedure.

preventLogon

If "preventLogon" or "preventLogon=y" is coded, then the user logon will be rejected during the job's runtime. If a user is already logged on, the execution of the job is delayed until the user logs off.
If "preventLogon=n" the user logon will not be rejected.

prompt=d.h

Specifies the time period in which the user will be prompted for the job's start.

The period's format is "*d.h*" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0). If the specified value falls below the valid minimum, the value is replaced by the minimum. If the specified value exceeds the valid maximum, then the value is replaced by the maximum. No warning is given.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

resolveQuery

Evaluate a possibly related query group before the job is generated.

rollback

Set the job linkage option to "Enable Transaction".

This parameter should only be coded, if the container is already in the batch processing mode or if it is switched to it by this command (parameter "transaction").

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

synchronized

The job linkage option is set to "Synchronized job execution".

transaction

The job linkage option is set to "Batch job execution".

userMessage

Specifies the custom administrator message for a job container.

Note: The first "uninstall" procedure of the items in question is used to generate the uninstall jobs for the items.

Example:

Suppose the job container "myContainer" has completed but some of the jobs has failed. It is not possible to clear the problems in an acceptable time. Therefore, the remaining parts from the failed execution on the targets systems should be removed. To recover the systems successfully the following requirements have to be met:

- Logged in user should be logged off before the jobs will be executed.
- No user should log in to the targets while the jobs are running.
- The jobs need a system restart after the last job has successfully completed.
- The clean up is provided by the default uninstall procedures.

The following command will schedule the recovery for the failed jobs:

```
cadsmcmd jobContainer action=recover name="myContainer" preaction=logoff
postaction=rebootAtEnd preventLogon
```

renew—Renew Job Container

This command renews all failed jobs of the specified job container. The specified parameter becomes active for all renewed jobs. The job options of the renewed jobs are modified on the basis of the related failed jobs (according to the parameters and their coded values). Options for the parameters that are not coded remain unchanged.

This command has the following format:

```
jobcontainer action=renew name=job_container_name
    [newName=new_container_name]
    [deliverytime="YYYY-MM-DD hh:mm" ]
    [attime="YYYY-MM-DD hh:mm" ]
    [after={exacttime|boottime}]
    [calendarname=delivery_calendar]
    [preaction={none|reboot|logoff}]
    [postaction={none|reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}]
    [promptUser[={y|n}]]
    [allowCancel[={y|n}]]
    [execTimedOut[={y|n}]]
    [prompt=days.hours]
    [offline[={y|n}]]
    [runAtShutdown[={y|n}]]
    [preventLogon[={y|n}]]
    [globalTime[={y|n}]]
    [nocalendar[={y|n}]]
    [resolveQuery[={y|n}]]
    [parameters=user_parameters]
    [jobTimeout=d.h]
    [userMessage]
```

after

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

allowCancel

If "allowCancel" or "allowCancel=y" is coded, the user will be granted to cancel the job's execution.
If "allowCancel=n" is coded, the user will not be granted.

This parameter should only be coded, if "promptUser" is already enabled or will be enabled with this call, otherwise this setting will be ignored. No warning is given.

atTime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

calendarName

Specifies the name of the calendar that controls the time when the evaluation can take place.

deliveryTime

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

execTimedOut

If "execTimedOut " or "execTimedOut=y" is coded, the job's execution will be automatically started when the user prompt times out.
If "execTimedOut=n", the job will not automatically be started.

This parameter should only be coded, if "promptUser" is already enabled or will be enabled with this call, otherwise this setting will be ignored. No warning is given.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

jobTimeout

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

name

Specifies the name of the job container to be renewed, in other words, all its failed jobs are renewed.

newName

Specifies the new name of the job container containing the renewed jobs. If this parameter is not coded, the old container is used.

noCalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

offline

If "offline" or "offline=y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.
If "offline=n" the connection will not be released.

parameters

Specifies the user parameters for the procedure.
Multiple user parameters must be separated by spaces inside double quotes.
If an empty string is coded, then no parameters are assumed.

postaction

Specifies the necessary actions that must take place after completion of the specified procedure. The following values are valid:

none

Performs no post-action.

reboot

Restarts the system after completion of the procedure.

logoff

Logs off the user after completion of the procedure.

rebootAtEnd

Restarts the system after completion of all the jobs of this container.

logoffAtEnd

Logs off the user after completion of all the jobs of this container.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

preaction

Specifies that necessary actions must take place before the specified procedure starts. The following values are valid:

none

Performs no pre-action.

reboot

Restarts the system before the start of the procedure.

logoff

Logs off the user before the start of the procedure.

preventLogon

If "preventLogon" or "preventLogon=y" is coded, then the user logon will be rejected during the job's runtime. If a user is already logged on, the execution of the job is delayed until the user logs off.
If "preventLogon=n" the user logon will not be rejected.

prompt=d.h

Specifies the time period in which the user will be prompted for the job's start.

The period's format is "*d.h*" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0). If the specified value falls below the valid minimum, the value is replaced by the minimum. If the specified value exceeds the valid maximum, then the value is replaced by the maximum. No warning is given.

This parameter should only be coded, if "promptUser" is already enabled or will be enabled with this call, otherwise this setting will be ignored. No warning is given.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

If the parameter is not coded, the default is given by the related procedure's job option value.

resolveQuery

Evaluate a possibly related query group before the job is generated.

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

userMessage

Specifies the custom administrator message for a renewed job container.

Example:

Assume that the job container "myContainer" has completed, but some of the jobs have failed. The problems have been cleared in the meantime, but to reschedule the jobs successfully, the following requirements have to be met:

- Logged in user should be logged out before the jobs will be executed.
- No user should log in to the targets while the jobs are running.
- The jobs need a system restart after the last job has successfully completed.

The following command will reschedule the failed jobs:

```
cadsmcmd jobContainer action=renew name="myContainer" preaction=logoff
postaction=rebootAtEnd preventLogon
```

resume—Resume Suspended Deliveries of Job Container

This action resumes the suspended deliveries of the specified job container. A delivery is an order to store a product at the staging library of a scalability server.

This action has the following format:

```
jobcontainer action=resume name=jobcontainername
```

name

Specifies the name of the job container to be resumed, in other words, its halted deliveries to scalability servers are resumed.

Example:

The following command resumes the deliveries of the job container "myContainer":

```
cadsmcmd jobContainer action=resume name="myContainer"
```

seal—Seal Job Container

This action seals the specified job container.

This action has the following format:

```
jobcontainer action=seal name=jobcontainername [evaluate]
```

name

Specifies the name of the job container to be sealed.

evaluate

The sealed job container will only be evaluated, but not activated.

If this parameter is not coded, the container will be evaluated and activated.

Important! The command is executed asynchronously!

Example:

Suppose the job container "myContainer" is populated now and should be sealed but not scheduled for execution yet.The following command seals the container:

```
cadsmcmd jobContainer action=seal name="myContainer" evaluate
```

setJcPriority—Set Job Container Priority

This action changes the priority of the job container regardless whether it is sealed or not.

This action has the following format:

```
jobcontainer action=setJcPriority
    name=job_container_name
    jcPriority={1,...,10}
```

name

Specifies the name of the job container of which the priority will be changed.

jcPriority

Specifies the new priority that will be assigned to the job container.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

showAttr—Show Attributes of Job Container

This action presents all attributes of a selected job container.

There are two types of presentations: (1) the target view (TV) with target-oriented presentation and (2) the job view with a job-oriented presentation.

This action has the following format:

```
jobcontainer action=showattr
    name=job_container_name
    [TV]
```

name

Specifies the name of the job container.

TV

Adjust the view of the list:

none specified

job view

TV

target view

Example:

To receive the lists shown in the description of the command "jobcontainer action=showattr" enter for the job oriented display:

```
cadsmcmd jobContainer action=showattr name="SD scalability servers [04.02.2006
15:42:39]"
```

The following job oriented information is shown:

```
------------------------------------------------------------------------------
Show attributes of the job container "SD scalabilityservers [04.02.2005 15:42:39]"
------------------------------------------------------------------------------
Job container name.....................: SD scalability servers [04.02.2005
15:42:39]
Job container properties...............: 20
.....................................Transaction
.....................................Cascade install
Job container priority:................: 5
Sealed.................................: Yes
```

```
Created at.............................: 2006-02-04 15:42
Changed at.............................: 2006-02-04 16:08
Number of contained objects............: 1
Job container state....................: failed
.......................................0 - jobs waiting
.......................................0 - jobs active
.......................................0 - jobs ok
.......................................1 - jobs failed
Job container renew state..............: in progress
Job name...............................: tstbase 1.0: Add to library
..Overall state........................: ERROR
.......................................0 - job targets waiting
.......................................0 - job targets active
.......................................0 - job targets ok
.......................................1 - job targets failed
..RenewState...........................: INPROGRESS
..Job order number.....................: 0
..Created at...........................: 2006-02-04 15:42
..Activate at..........................: 2006-02-04 15:42
..Status message.......................: OK
..Target...............................: KKKKK01A
....Task...............................: deliver
....Job state..........................: DELIVERY_ERROR
....RenewState.........................: DELIVERY_ORDERED
....Task...............................: deliver
....Started at.........................: 2006-02-04 15:42
....Completed at.......................: 2006-02-04 15:46
....Created at.........................: 2006-02-04 15:42
....Data sent in per cent..............: 100
....Status message.....................: DTS error: DTS transfer error: 447 E  E
.......................................+ (447) Failed calling external file fil
.......................................+ ter <sddtaflt>, internal code=<0>, ext
.......................................+ ernal code=<4>, errmsg=<>.
....Ordered by.........................: kkkkk01
....Type...............................: mandatory
....Comment............................:
```

and for the target oriented display:

```
cadsmcmd jobContainer action=showattr name="SD scalability servers [04.02.2006
15:42:39]" TV
```

The following target oriented information is shown:

```
--------------------------------------------------------------------------------
Show attributes of the job container "SD scalability servers [04.02.2005 15:42:39]"
(TV)
--------------------------------------------------------------------------------
Job container name.....................: SD scalability servers [04.02.2005
15:42:39]
Job container properties...............: 20
.......................................Transaction
.......................................Cascade install
Sealed.................................: Yes
Created at.............................: 2006-02-04 15:42
Changed at.............................: 2006-02-04 16:08
Number of contained objects............: 1
Job container state....................: failed
.......................................0 - jobs waiting
.......................................0 - jobs active
.......................................0 - jobs ok
.......................................1 - jobs failed
Job container renew state..............: in progress

Target.................................: KKKKK01A
..Job overall state....................: ERROR
.......................................0 - job targets waiting
.......................................0 - job targets active
.......................................0 - job targets ok
.......................................1 - job targets failed
..RenewState...........................: INPROGRESS
..Job name.............................: tstbase 1.0: Add to library
....Software...........................: tstbase 1.0
....Procedure..........................: delivery proc
....Install procedure..................: delivery proc
....State..............................: DELIVERY_ERROR
....RenewState.........................: DELIVERING
....Task...............................: deliver
....Started at.........................: 2006-02-04 15:42
....Completed at.......................: 2006-02-04 15:46
....Created at.........................: 2006-02-04 15:42
....Data sent in per cent..............: 100
```

```
....Status message.....................: DTS error: DTS transfer error: 447 E  E
.......................................+ (447) Failed calling external file fil
.......................................+ ter <sddtaflt>, internal code=<0>, ext
.......................................+ ernal code=<4>, errmsg=<>.
....Ordered by........................: kkkkk01
....Type..............................: mandatory
....Comment...........................:
```

unseal—Unseal Job Container

This action unseals the specified job container.

This action has the following format:

`jobcontainer action=unseal name=`*`jobcontainername`*

name

Specifies the name of the job container to be unsealed.

Example:

Suppose the job container "myContainer" has to be modified and therefore has to be unsealed again. The following command will unseal the container:

```
cadsmcmd jobContainer action=unseal name="myContainer"
```

Job Management Commands

This section contains the following topics:

addJob—Add Job to Job Container

This command adds a job to an existing job container. You must not seal the job container.

This command has the following format:

```
jobcontainer action=addjob
    name=job_container_name
    {item=item_name version=item_version
    procedure=procedure_name
    {[task=install]
    |task={configure|activate|uninstall}
    [installedWith=install_procedure]}
    | group=group_name [procedures=file]}
    {{{compgrp=computer_group_name | ccompgrp=(computer_group_name sep groupScope)}
    [sep=separator_sign]
    |{computer=computer_name}}
    [deliverytime="YYYY-MM-DD hh:mm" ]
    [attime="YYYY-MM-DD hh:mm" ]
    [{preaction={none|reboot|logoff}|bootbefore}]
    [{postaction={none|reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}|boota
    fter}]
    [promptUser[={y|n}]]
    [allowCancel[={y|n}]]
    [execTimedOut[={y|n}]]
    [offline[={y|n}]]
    [runAtShutdown[={y|n}]]
    [preventLogon[={y|n}]]
    after={exacttime|boottime}
    [calendarname=delivery_calendar]
    [globalTime=y|n]
    [nocalendar[={y|n}]]
    [resolveQuery[={y|n}]]
    [stagingServer[={y|n}]]
    [parameters=user_parameters]
    [prompt=days.hours]]
    [jobTimeOut=days.hours]
    [reinstall[={y|n}]]
    [userJobMessage]
```

after

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

allowCancel

If "allowCancel" or "allowCancel =y" is coded, the user will be granted to cancel the job's execution.

If "allowCancel =n" is coded, the user will not be granted.

If the parameter is not coded, the default is given by the related procedure's job option value.

atTime

Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

calendarname

Specifies the name of the calendar that controls the time when the evaluation can take place.

ccompgrp

Specifies the computer group to be addressed as a pair of group name and scope. By default, name and scope are separated by a dot ("."), but an alternate separator can be specified with the "sep" option. The separator should **not** be used as part of the names.

For the group scope only the following values are valid.

local

The group is created at the local domain manager,

global

The group has been created at the enterprise manager and may be replicated.

The "ccompgrp" can be coded with the "compgrp" in one call.

compgrp

Specifies the name of a computer group on which the specified product will be processed.

The parameter can be coded more than once to address more than one computer group.

The parameter must not be coded together with the computer parameter.

computer

Specifies the name of the computer on which the specified product will be processed.

The parameter can be coded more than once to address more than one computer.

The parameter must not be coded together with the compgrp parameter.

deliveryTime

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

execTimedOut

If "execTimedOut " or "execTimedOut=y" is coded, the job's execution will be automatically started when the user prompt times out.

If "execTimedOut=n", the job will not automatically be started.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

group

Specifies the name of a software or procedure group the jobs will be generated for.

The parameter must not be coded together with the item parameter.

installedWith

Specifies the name of the install procedure.

This parameter is mandatory for the tasks configure, activate, and uninstall. Otherwise it is ignored.

item

Specifies the name of the item to be processed.

This parameter must not be coded together with the group parameter.

jobTimeOut

Specifies the expiration date of the job in form of a period.

The period is specified in the format "d.h" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0).

If the specified value falls below the valid minimum, the value is replaced by the minimum.

If the specified value exceeds the valid maximum, then the value is replaced by the maximum.

No warning is given.

name

Specifies the name of the job container to which the job will be added.

noCalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

offline

If "offline" or "offline =y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.

If "offline =n" the connection will not be released.

parameters

Specifies the user parameters for the procedure.

Multiple user parameters must be space separated inside double quotes.

If an empty string is coded, then no parameters are assumed any longer.

postaction

Specifies that necessary actions must take place after the specified procedure has terminated. The following values are valid:

none

Performs no post-action.

reboot

Restarts the system after completion of the procedure.

logoff

Logs off the user after completion of the procedure.

rebootAtEnd

Restarts the system after completion of all jobs of this container.

logoffAtEnd

Logs off the user after all completion of all jobs of this container.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

preaction

Specifies that necessary actions must take place before the specified procedure starts. The following values are valid:

none

Performs no pre-action.

reboot

Restarts the system before the start of the procedure.

logoff

Logs off the user before the start of the procedure.

preventLogon

If "preventLogon" or "preventLogon =y" is coded, then the user logon will be rejected during the job's runtime. If a user is already logged on, the execution of the job is delayed until the user logs off.

If "preventLogon =n" the user logon will not be rejected.

procedure

Specifies the name of a procedure

procedures

Specifies the name of the file defining the procedures to be used for the software group and / or to define individual job options for the procedures (software and procedure groups).

For the content of this procedures file, see Procedures File Format (see page 164)

prompt

Specifies the time period in which the user will be prompted for the job's start.

The period’s format is "d.h" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0). If the specified value falls below the valid minimum, the value is replaced by the minimum. If the specified value exceeds the valid maximum, then the value is replaced by the maximum. No warning is given.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

reinstall

Optional. Default: “reinstall=n”

If “reinstall” or “reinstall=y” is coded then a possible installation record related to the order is removed before the job is evaluated by the domain manager.

If “reinstall=n” is coded then such records will remain.

Note: Option only applies to install jobs. In any other cases it is ignored.

resolveQuery

Evaluate a possibly related query group before the job is generated.

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.

If "runAtShutdown=n", the job will start at any other time too.

If the parameter is not coded, the default is given by the related procedure's job option value.

This parameter is for Unix targets only.

sep

Specifies an alternate separator used with the "ccompgroup" option. The separator should not be part of the group name code with the "ccompgroup" options.

stagingServer

Deliver the related product also to the staging libraries of the scalability servers of the target computers.

This parameter applies to procedures of task type "install" only!

task

Specifies the type of task to be performed by the item procedure.

The following types are valid:

activate

The procedure activates the item.

configure

The procedure configures the item.

install

The procedure installs the item.

uninstall

The procedure uninstalls the item.

If the parameter is not coded, install is assumed.

version

Specifies the version of the item to be processed.

userJobMessage

Specifies the custom administrator message for a job in a job container.

Note: You should be aware of the following:

- The parameters "allowCancel," "execTimedOut," and "prompt" are valid if the "promptUser" option is already set or is set by this command using the "promptUser" parameter. If neither the "promptUser" option of a job nor the "promptUser" parameter of the command is set, these parameters will be ignored, and no warning is given.
- If group is coded without procedures and the group specified is a software group then installation jobs are generated for the group members where the first install procedure according to the members numbering is taken as job procedure.

Procedures file format

The procedures file consists of entries of the following syntax:

```
item = item
version = version
procedure = procedure
task={install|configure|activate|uninstall}
[deliverytime="YYYY-MM-DD hh:mm" ]
[attime="YYYY-MM-DD hh:mm" ]
[after={exacttime|boottime}]
[calendarname=deliverycalendar]
[preaction={none|reboot|logoff}]
[postaction={none|reboot|logoff|rebootAtEnd|logoffAtEnd}]
[promptUser={y|n}]
[allowCancel={y|n}]
[execTimedOut={y|n}]
[prompt=days.hours]
[offline={y|n}]
[RunAtShutdown={y|n}]
[PreventLogon={y|n}]
[GlobalTime={y|n}]
[nocalendar={y|n}]
[ResolveQuery={y|n}]
[Parameters=userparameter]
[stagingserver={y|n}]
[JobTimeout=d.h]
```

item

Name of the group item this entry is related to

version

Version for the group item this entry is related to

procedure

Specifies the item procedure to be performed

task

Specifies the procedure type.

The following types are valid:

install

The procedure installs the item.

configure

The procedure configures the item

activate

The procedure activates the item.

uninstall

The procedure uninstalls the item.

If the parameter is not coded, install is assumed.

The parameters "item", "version", "procedure", and "task" have to be coded at each line as the first parameters and they have to be coded in the order as in the syntax definition previously shown. The remaining parameter can be coded in any order.

The meaning of the other parameters corresponds to the meaning of the command parameters shown in cadsmcmd jobcontainer action=addjob.

The procedure options for a job procedure are taken from the settings in this list.

If for a procedure an option is not coded in the related list entry then it is taken from the settings on command layer.

If it is not coded there then it is taken from the procedure options registered.

Example:

Consider the unsealed job container "myContainer". Assume that the item "prod 1" of version 1.0 should be installed at KKKK01A and KKKK01B using the install procedure "inst". The user at the targets should be prompted for installation. The install order should be placed into the job container "myContainer".

The following command will provide the required job:

```
cadsmcmd jobContainer action=addJob name=myContainer item="prod 1" version=1.0
procedure=inst task=install computer=KKKK01A computer=KKKK01B promptUser
after=exacttime
```

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM
Generate job "prod 1 1.0:inst": OK.

SDCMD<A000000>: OK
```

addDelivery—Add Delivery to Job Container

This action adds a job to the specified job container that delivers the specified software products to the addressed scalability servers, in other words, it stores them to the software libraries of the scalability servers. The specified job container has to be unsealed.

This action has the following format:

```
jobcontainer action=addDelivery
    name=job_container_name
    {item=item_name
    version=item_version| group=software_group_name}
    {{{compgrp=computer_group_name | ccompgrp=(computer_group_name sep groupScope)}
```

```
[sep=separator_sign]
|{computer=computer_name}}
[calendar_name=delivery_calendar]
[deliverytime="YYYY-MM-DD hh:mm" ]
[jobTimeOut=d.h]
```

calendarname

Specifies the name of the calendar that controls the time when the evaluation can take place.

ccompgrp

Specifies the computer group to be addressed as a pair of group name and scope. By default, name and scope are separated by a dot ("."), but an alternate separator can be specified with the "sep" option. The separator should **not** be used as part of the names.

For the group scope only the following values are valid.

local

The group is created at the local domain manager,

global

The group has been created at the enterprise manager and may be replicated.

The "ccompgrp" can be coded with the "compgrp" in one call.

compgrp

Specifies the name of a server group on which the specified product will be processed.

The parameter can be coded more than once to address more than one server group.

This parameter must not be coded with the computer parameter.

computer

Specifies the name of the scalability server on which the specified product will be processed.

The parameter can be coded more than once to address more than one scalability server.

This parameter must not be coded with the compgrp parameter.

deliverytime

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

group

Specifies the name of a software group: Its members will be delivered to the specified scalability servers.

This parameter should not be coded with the item parameter.

item

Specifies the name of the item to be processed.

This parameter should not be coded with the group parameter.

jobTimeout

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

name

Specifies the name of the job container to which the delivery to the scalability servers will be added.

sep

Specifies an alternate separator used with the "ccompgroup" option. The separator should not be part of the group name code with the "ccompgroup" options.

version

Specifies the version of the item to be processed.

Example:

Consider the unsealed job container "myContainer". Assume that the item "prod 2" of Version 1.0 should be delivered to the scalability server KKKKK01B. The delivery order should be placed into the job container "myContainer". The following command will provide the required job:

```
cadsmcmd jobContainer action=addDelivery name="myContainer" item="prod 2"
version=1.0 computer=KKKKK01B


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM

SDCMD<A000000>: OK
```

addUndelivery—Add Undelivery to Job Container

This action adds a job to the specified job container that removes the specified software products from the addressed scalability servers. The specified job container has to be unsealed.

This action has the following format:

```
jobcontainer action=addUnDelivery
    name=job_container_name
    {item=item_name
    version=item_version| group=software_group_name}
    {{compgrp=computer_group_name | ccompgrp=(computer_group_name sep groupScope)}
    [sep=separator_sign]
    |{computer=computer_name}}
    [deliverytime="YYYY-MM-DD hh:mm" ]
    [calendar_name=delivery_calendar]
    [jobTimeOut=d.h]
```

calendarname

Specifies the name of the calendar that controls the time when the evaluation can take place.

ccompgrp

Specifies the computer group to be addressed as a pair of group name and scope. By default, name and scope are separated by a dot ("."), but an alternate separator can be specified with the "sep" option. The separator should **not** be used as part of the names.

For the group scope only the following values are valid.

local

The group is created at the local domain manager,

global

The group has been created at the enterprise manager and may be replicated.

The "ccompgrp" can be coded with the "compgrp" in one call.

compgrp

Specifies the name of a server group on which the specified product will be processed.

The parameter can be coded more than once to address more than one server group.

The parameter must not be coded with the computer parameter.

computer

Specifies the name of the scalability server on which the specified product will be processed.

The parameter can be coded more than once to address more than one scalability server.

This parameter must not be coded with the compgrp parameter.

deliverytime

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

group

Specifies the name of a software group: Its members will be removed from the specified scalability servers.

This parameter should not be coded with the item parameter.

item

Specifies the name of the item to be processed.

This parameter must not be coded with the group parameter.

jobTimeout

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

name

Specifies the name of the job container to which the un-delivery to the scalability servers will be added.

sep

Specifies an alternate separator used with the "ccompgroup" option. The separator should not be part of the group name code with the "ccompgroup" options.

version

Specifies the version of the item to be processed.

listJobs—List Jobs

This action lists all jobs of the specified job container. A filter can be specified to confine the amount of jobs listed.

This action has the following format:

```
jobcontainer action=listJobs
    name=job_container_name
    {[{filter=filter|filterfile=file_name}]
    [TV] | TV alt}
```

alt

Provides an alternate - more detailed - list in target view.

An alternate list is provided based on the item name and version only.

In case of coding this parameter the value of the item and version parameter might contain database supported wild cards, for example,

"%"

for an arbitrary, even empty sequence of characters

"_"

for an arbitrary single character

For Microsoft SQL Server also the square brackets ("[","]") have a special meaning. This has to be taken into account when coding the name.

name

Specifies the name of the job container to be listed.

filter

Specifies the expression to confine the amount of information listed.

Attribute	**Type**
Job name	String
Overall state	{ WAITING | INPROGRESS | OK | ERROR | BUILDING | REMOVING | NOT_AVAILABLE | CREATED | OBSOLETE}
Software	String
Procedure	String
Deliver at	DateTime
Activate at	DateTime
Status message	String
Job state	{UNDEFINED | WAITING | DELIVERY_ORDERED | DELIVERING | DELIVERY_OK | DELIVERY_ERROR | ALREADY_DELIVERED | EXECUTION_ORDERED | EXECUTING | EXECUTION_OK | EXECUTION_ERROR | UNDELIVERY_ORDERED | UNDELIVERING | UNDELIVERY_OK | UNDELIVERY_ERROR | ALREADY_INSTALLED | MANIPULATION_NOT_ALLOWED}

Attribute	Type
Job task	{INSTALL | CONFIGURE | ACTIVATE | UNINSTALL | DELIVER | UNDELIVER}
Pre action	{NO ACTION | REBOOT | LOGOFF}
Post action	{NO ACTION | REBOOT | LOGOFF | REBOOT AFTER LAST JOB | LOGOFF AFTER LAST JOB}
Delivery calendar	String
User parameters	String

The atomic attributes correspond to those shown at showAttrJobs or showJobDetails.

filterfile

name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command.

TV

Adjust the view of the list:

none specified

job view

TV

target view

Example:

To list the contents of the container “e.g.“ the following command should be entered:

```
cadsmcmd jobContainer action=listJobs name=e.g.
```

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "wwwww01b" as user "<default user>" ...OK.
```

```
--------------------------------------------------------------------------------
Job list of job container “e.g.“
--------------------------------------------------------------------------------
Job name.............................: prod 1 1.0:inst
..Task...............................: install
..Overall state......................: WAITING
..Job order number...................: 1

Job name.............................: prod 3 1.0:inst
..Task...............................: install
..Overall state......................: WAITING
..Job order number...................: 2

Job name.............................: prod 2 1.0:delivery proc
..Task...............................: deliver
..Overall state......................: WAITING
..Job order number...................: 3

--------------------------------------------------------------------------------
Number of jobs read: 3
Number of jobs listed: 3
sdcmd<a000000>: OK
```

The previous list is shown, when the job container is not sealed. After sealing, you may receive the following list:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "wwwww01b" as user "<default user>" ...OK.

--------------------------------------------------------------------------------
Job list of job container “e.g.“
--------------------------------------------------------------------------------
Job name.............................: prod 1 1.0:inst
..Task...............................: install
..Overall state......................: OK
..Job order number...................: 1
..Target.............................: KKKKK01A
....Job state........................: EXECUTION_OK
..Target.............................: KKKKK01B
....Job state........................: EXECUTION_OK
```

```
Job name.............................: prod 3 1.0:inst
..Task...............................: install
..Overall state......................: ERROR
..Job order number...................: 2
..Target.............................: KKKKK01A
....Job state........................: EXECUTION_OK
..Target.............................: KKKKK01B
....Job state........................: EXECUTION_ERROR

Job name.............................: prod 2 1.0:delivery proc
..Task...............................: deliver
..Overall state......................: OK
..Job order number...................: 3
..Target.............................: KKKKK01B
....Job state........................: DELIVERY_OK
--------------------------------------------------------------------------------
Number of jobs read: 3
Number of jobs listed: 3
sdcmd<a000000>: OK
```

This list is a job-oriented list. To retrieve a target oriented list enter the following command:

```
cadsmcmd jobContainer action=listJobs name="e.g." TV


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "wwwww01b" as user "<default user>" ...OK.

--------------------------------------------------------------------------------
Job list of job container "e.g." (TV)
--------------------------------------------------------------------------------
Target...............................: KKKKK01A
..Job overall state..................: OK
..Job name...........................: prod 1 1.0:inst
....Task.............................: install
....Job state........................: EXECUTION_OK
..Job name...........................: prod 3 1.0:inst
....Task.............................: install
....Job state........................: EXECUTION_OK
```

```
Target.............................: KKKKK01B
..Job overall state................: ERROR
..Job name.........................: prod 1 1.0:inst
....Task...........................: install
....Job state......................: EXECUTION_OK
..Job name.........................: prod 3 1.0:inst
....Task...........................: install
....Job state......................: EXECUTION_ERROR
..Job name.........................: prod 2 1.0:delivery proc
....Task...........................: deliver
....Job state......................: DELIVERY_OK

--------------------------------------------------------------------------------
Number of jobs read: 5
Number of jobs listed: 5
sdcmd<A000000>: OK
```

When the following is invoked

```
cadsmcmd jobContainer action=listJobs name="e.g." TV alt
then the returned list might look as follows:


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager: mgr_1
Domain: mgr-1
Domain type: Domain
Supporting: CO CCNF USD OSIM
```

```
--------------------------------------------------------------------------------
Job list of items for job container "e.g."
--------------------------------------------------------------------------------
Target.............................: KKKKK01
..Job name.........................: prod 1 1.0:inst
..Job identifier...................: 27B1661D-ED2D-DB01-B402-000F1F7090E6
..Item name........................: prod 1
..Item version.....................: 1.0
..Procedure name...................: inst
..Task.............................: install
..User parameters..................:
..Job state........................: EXECUTION_ERROR
..Error code.......................: 228001
..Error message....................: 1
..Creation date....................: 2006-08-17
..Creation time....................: 14:37
..Activation date..................: 2006-08-17
..Activation time..................: 14:36
..Completion date..................: 2006-08-17
..Completion time..................: 14:37
```

To list Jobs that are with overall state 'ok', enter the following command:

```
cadsmcmd jobcontainer action=listjobs name=e.g. filter="Overall state=OK"
```

The returned list is as follows:

```
Job name...........................: prod 1 1.0:inst
..Task.............................: install
..Overall state....................: OK
..Job order number.................: 1
..Target...........................: KKKKK01A
....Job state......................: EXECUTION_OK
..Target...........................: KKKKK01B
....Job state......................: EXECUTION_OK
Job name...........................: prod 2 1.0:delivery proc
..Task.............................: deliver
..Overall state....................: OK
..Job order number.................: 3
..Target...........................: KKKKK01B
....Job state......................: DELIVERY_OK

Number of jobs read: 3
Number of jobs listed: 2
sdcmd<a000000>: OK
```

listFailedJobs—List Failed Jobs

This action lists all failed jobs of the specified job container. A filter can be specified to confine the amount of failed jobs listed.

This action has the following format:

```
jobcontainer action=listFailedJobs
    name=job_container_name
    {[{filter=filter|filterfile=file_name}]
    [TV] | TV alt}
```

alt

Provides an alternate - more detailed - list in target view.

An alternate list is provided based on the item name and version only.

In case of coding this parameter the value of the item and version parameter might contain database supported wild cards, for example,

"%"

for an arbitrary, even empty sequence of characters

"_"

for an arbitrary single character

For Microsoft SQL Server also the square brackets ("[","]") have a special meaning. This has to be taken into account when coding the name.

name

Specifies the name of the job container to be listed, however, only failed jobs will be listed.

filter

Specifies the expression to confine the amount of information listed.

The atomic expressions for building filter expressions correspond to those of the listjobs.

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command..

TV

Adjust the view of the list:

none specified

job view

TV

target view

The information listed is of the same format as for "listJobs".

Example:

To list the failed jobs of the container "myContainer" the following command should be entered:

```
cadsmcmd jobContainer action=listFailedJobs name="myContainer"


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM
```

```
--------------------------------------------------------------------------------
Job list of job container "myContainer"
--------------------------------------------------------------------------------
Job name.............................: prod 2 1.0:inst
..Task...............................: install
..Overall state......................: ERROR
..Job order number...................: 2
..Target.............................: KKKKK01A
....Job state........................: EXECUTION_OK
..Target.............................: KKKKK01B
....Job state........................: EXECUTION_ERROR
Job name.............................: prod 3 1.0:inst
..Task...............................: configure
..Overall state......................: ERROR
..Job order number...................: 3
..Target.............................: KKKKK01A
....Job state........................: EXECUTION_ERROR
..Target.............................: KKKKK01B
....Job state........................: EXECUTION_ERROR

--------------------------------------------------------------------------------
Number of jobs read: 2
Number of jobs listed: 2
SDCMD<A000000>: OK
```

To see a target -oriented list, enter:

```
cadsmcmd jobContainer action=listFailedJobs name=myContainer TV

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "wwwww01b" as user "<default user>" ...OK.

--------------------------------------------------------------------------------
Job list of job container "myContainer" (TV)
--------------------------------------------------------------------------------
Target...............................: KKKKK01B
..Job overall state..................: ERROR
..Job name...........................: prod 3 1.0:inst
....Task.............................: install
....Job state........................: EXECUTION_ERROR
```

```
--------------------------------------------------------------------------------
Number of jobs read: 1
Number of jobs listed: 1
SDCMD<A000000>: OK
```

When the following is invoked

```
cadsmcmd jobContainer action=listFailedJobs name="e.g." TV alt
```

then the returned list might look as follows:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager: mgr_1
Domain: mgr-1
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
Job list of items for job container "e.g."
--------------------------------------------------------------------------------
Target.................................: KKKKK01
..Job name.............................: prod 1 1.0:inst
..Job identifier.......................: 27B1661D-ED2D-DB01-B402-000F1F7090E6
..Item name............................: prod 1
..Item version.........................: 1.0
..Procedure name.......................: inst
..Task.................................: install
..User parameters......................:
..Job state............................: EXECUTION_ERROR
..Error code...........................: 228001
..Error message........................: 1
..Creation date........................: 2006-08-17
..Creation time........................: 14:37
..Activation date......................: 2006-08-17
..Activation time......................: 14:36
..Completion date......................: 2006-08-17
..Completion time......................: 14:37
```

To list jobs failed during installation using filters, enter the following command:

```
cadsmcmd jobContainer action=listFailedJobs name=myContainer filter=”Job
task=INSTALL”
```

The output is as follows:

```
--------------------------------------------------------------------------
Job list of job container "myContainer"
--------------------------------------------------------------------------------
Job name.............................: prod 2 1.0:inst
..Task...............................: install
..Overall state......................: ERROR
..Job order number...................: 2
..Target.............................: KKKKK01A
....Job state........................: EXECUTION_OK
..Target.............................: KKKKK01B
....Job state........................: EXECUTION_ERROR

--------------------------------------------------------------------------------
Number of jobs read: 2
Number of jobs listed: 1
SDCMD<A000000>: OK
```

modfiyJob—Modify Job

This command lets you modify a job of the specified job container. You must unseal the job container.

The parameters are related to job options. If a parameter is not coded, the related option remains unchanged. You must code the parameter "stagingServer" with install tasks; otherwise, it will cause an error.

This command has the following format:

```
jobcontainer action=modifyjob
    name=job_container_name
    jobName=job_name [newJobName=new_job_name]
    [deliveryTime="YYYY-MM-DD hh:mm"]
```

```
[atTime="YYYY-MM-DD hh:mm"]
[after={exacttime|boottime}]
[calendarname=delivery_calendar]
[preaction={none|reboot|logoff}]
[postaction={none|reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}]
[promptUser[={y|n}]]
[allowCancel[={y|n}]]
[execTimedOut[={y|n}]]
[prompt=days.hours]
[offline[={y|n}]]
[runAtShutdown[={y|n}]]
[preventLogon[={y|n}]]
[globalTime[={y|n}]]
[nocalendar[={y|n}]]
[resolveQuery[={y|n}]]
[stagingServer[={y|n}]]
[parameters=user_parameters]
[jobTimeout=d.h]
[reinstall[={y|n}]]
[userJobMessage]
```

after

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

allowCancel

If "allowCancel" or "allowCancel=y" is coded, the user will be granted to cancel the job's execution.
If "allowCancel=n" is coded, the user will not be granted.

This parameter should only be coded, if "promptUser" is already enabled or will be enabled with this call, otherwise this setting will be ignored. No warning is given.

atTime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

calendarname

Specifies the name of the calendar that controls the time when the evaluation can take place.

deliverytime

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

execTimedOut

If "execTimedOut " or "execTimedOut=y" is coded, the job's execution will be automatically started when the user prompt times out.
If "execTimedOut=n", the job will not automatically be started.

This parameter should only be coded, if "promptUser" is already enabled or will be enabled with this call, otherwise this setting will be ignored. No warning is given.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

jobName

Specifies the name of the job to be modified.

jobTimeout

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

name

Specifies the name of the job container to which the job to be modified belongs.

newJobName

Specifies the new job name.

noCalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

offline

If "offline" or "offline=y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.
If "offline=n" the connection will not be released.

parameters

Specifies the user parameters for the procedure.
Multiple user parameters must be separated by spaces inside double quotes.
If an empty string is coded, then no parameters are assumed.

postaction

Specifies the necessary actions that must take place after completion of the specified procedure. The following values are valid:

none

Performs no post-action.

reboot

Restarts the system after completion of the procedure.

logoff

Logs off the user after completion of the procedure.

rebootAtEnd

Restarts the system after completion of all the jobs of this container.

logoffAtEnd

Logs off the user after completion of all the jobs of this container.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

preaction

Specifies that necessary actions must take place before the specified procedure starts. The following values are valid:

none

Performs no pre-action.

reboot

Restarts the system before the start of the procedure.

logoff

Logs off the user before the start of the procedure.

preventLogon

If "preventLogon" or "preventLogon=y" is coded, then the user logon will be rejected during the job's runtime. If a user is already logged on, the execution of the job is delayed until the user logs off.
If "preventLogon=n" the user logon will not be rejected.

prompt=d.h

Specifies the time period in which the user will be prompted for the job's start.

The period's format is "*d.h*" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0). If the specified value falls below the valid minimum, the value is replaced by the minimum. If the specified value exceeds the valid maximum, then the value is replaced by the maximum. No warning is given.

This parameter should only be coded, if "promptUser" is already enabled or will be enabled with this call, otherwise this setting will be ignored. No warning is given.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

reinstall

The "reinstall" option is optional, no default.

If not coded the reinstall settings for the addressed jobs are not changed otherwise they are changed according to the coding

Note: Option only applies to install jobs. In any other cases it is ignored.

resolveQuery

Evaluate a possibly related query group before the job is generated.

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

stagingServer

Deliver the related product also to the staging libraries of the scalability servers of the target computers.

This parameter applies to procedures of task type "install" only!

This parameter should be only coded with procedures of task type "install". Otherwise an error will be reported.

userJobMessage

Modifies the custom administrator message of a job in a job container.

Example:

Assume the container "myContainer" is unsealed and the job "prod 3 1.0:inst" should be changed in such a way that the product is automatically delivered to scalability servers associated with the assigned targets. The following command will change the job to meet this requirement:

```
cadsmcmd jobContainer action=modifyJob name="myContainer" jobName="prod 3 1.0:inst"
stagingServer=y
```

recoverJob—Recover Job from Failed Job Container

This command recovers a failed job from the specified job container, that is, it runs an uninstall procedure to clean up the failed job's residues.

The job options for the recover job are modified on the basis of the referred job according to the parameters and their coded values. The options belonging to parameters not coded remain unchanged for the recover job. The same holds for the options of the recovery container. The defaults are given by the source container (parameter "name").

This command has the following format:

```
jobcontainer action=recoverJob
    name=job_container_name
    jobName=job_name
    [newName=new_container_name]
    [newJobName=new_job_name]
    [procedure=procedure_name]
```

```
[{transaction|synchronized|noLinkage}]
[rollback[={y|n}]]
[cascade[={y|n}]]
[deliveryTime="YYYY-MM-DD hh:mm"]
[atTime="YYYY-MM-DD hh:mm"]
[after={exacttime|boottime}]
[calendarname=delivery_calendar]
[preaction={none|reboot|logoff}]
[postaction={none|reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}]
[promptUser[={y|n}]]
[allowCancel[={y|n}]]
[execTimedOut[={y|n}]]
[prompt=days.hours]
[offline[={y|n}]]
[runAtShutdown[={y|n}]]
[preventLogon[={y|n}]]
[globalTime[={y|n}]]
[nocalendar[={y|n}]]
[resolveQuery[={y|n}]]
[stagingServer[={y|n}]]
[parameters=user_parameters]
[jobTimeout=d.h]
[userJobMessage]
[userMessage]
```

after

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

allowCancel

If "allowCancel" or "allowCancel=y" is coded, the user will be granted to cancel the job's execution.
If "allowCancel=n" is coded, the user will not be granted.

This parameter should only be coded, if "promptUser" is already enabled or will be enabled with this call, otherwise this setting will be ignored. No warning is given.

atTime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

calendarName

Specifies the name of the calendar that controls the time when the evaluation can take place.

cascade

If "cascade" or "cascade=y" is coded, the job linkage of the container is switched to resolution of dependencies into install cascades.
If "cascade=n" is coded, the job linkage is switched to no resolution.

This parameter should only be coded, if the container is already in the batch processing mode or if it is switched to it by this command (parameter "transaction").

deliveryTime

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

execTimedOut

If "execTimedOut " or "execTimedOut=y" is coded, the job's execution will be automatically started when the user prompt times out.
If "execTimedOut=n", the job will not automatically be started.

This parameter should only be coded, if "promptUser" is already enabled or will be enabled with this call, otherwise this setting will be ignored. No warning is given.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

jobName

Specifies the name of the job to be recovered.

For all failed target systems, an "uninstall" procedure is launched to clean the system.

jobTimeout

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

name

Specifies the name of the job container from which a job will be recovered.

newJobName

Specifies the new job name, that is, the name of the recover job.

If not coded a default job name of the form

```
"<item name> <item version>:<selected uninstall procedure name>"
```

is generated.

newName

Specifies the new name of the job container, which will contain the recover jobs.

If this parameter is not coded a default container name of the form

```
"Recover[<timestamp>] <name of the job to be recovered> [<timestamp>]"
```

is generated.

noCalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

noLinkage

Run the job independently of the other jobs in the container.

offline

If "offline" or "offline=y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.
If "offline=n" the connection will not be released.

parameters

Specifies the user parameters for the procedure.
Multiple user parameters must be separated by spaces inside double quotes.
If an empty string is coded, then no parameters are assumed.

postaction

Specifies the necessary actions that must take place after completion of the specified procedure. The following values are valid:

none

Performs no post-action.

reboot

Restarts the system after completion of the procedure.

logoff

Logs off the user after completion of the procedure.

rebootAtEnd

Restarts the system after completion of all the jobs of this container.

logoffAtEnd

Logs off the user after completion of all the jobs of this container.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

If the parameter is not coded, the default is given by the related procedure's job option value.

preaction

Specifies that necessary actions must take place before the specified procedure starts. The following values are valid:

none

Performs no pre-action.

reboot

Restarts the system before the start of the procedure.

logoff

Logs off the user before the start of the procedure.

If the parameter is not coded, the default is given by the related procedure's job option value.

preventLogon

If "preventLogon" or "preventLogon=y" is coded, then the user logon will be rejected during the job's runtime. If a user is already logged on, the execution of the job is delayed until the user logs off.
If "preventLogon=n" the user logon will not be rejected.

procedure

Specifies the name of the "uninstall" procedure to be performed.

prompt=d.h

Specifies the time period in which the user will be prompted for the job's start.

The period's format is "*d.h*" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0). If the specified value falls below the valid minimum, the value is replaced by the minimum. If the specified value exceeds the valid maximum, then the value is replaced by the maximum. No warning is given.

This parameter should only be coded, if "promptUser" is already enabled or will be enabled with this call, otherwise this setting will be ignored. No warning is given.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

If the parameter is not coded, the default is given by the related procedure's job option value.

resolveQuery

Evaluate a possibly related query group before the job is generated.

rollback

Set the job linkage option to "Enable Transaction".

This parameter should only be coded, if the container is already in the batch processing mode or if it is switched to it by this command (parameter "transaction").

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

synchronized

The job linkage option is set to "Synchronized job execution".

transaction

The job linkage option is set to "Batch job execution".

This is the default.

userJobMessage

Specifies the custom administrator message for a job in a job container.

userMessage

Specifies the custom administrator message for a job container.

Note: The first "uninstall" procedure of the items in question is used to generate the uninstall jobs for the items.

Example:

Suppose the job container "myContainer“ has completed but some of the jobs have failed. It is not possible to clear the problems in an acceptable time. Therefore, the remaining parts from the failed execution on the targets systems should be removed. To recover the systems successfully the following requirements have to be met:

- Logged in user should be logged off before the jobs will be executed.
- No user should log in to the targets while the jobs are running.
- The jobs need a system restart after the last job has successfully completed.
- The clean up is provided by the default uninstall procedures.

The following command will schedule the recovery for the failed jobs:

```
cadsmcmd jobContainer action=recover name="myContainer“ preaction=logoff
postaction=rebootAtEnd preventLogon
```

removeJob—Remove Job from Job Container

This action removes a job from a job container.

This action has the following format:

```
jobcontainer action=removeJob
    name=jobcontainername
    jobName=jobname
```

name

Specifies the name of the job container from which the job will be removed.

jobName

Specifies the name of the job to be removed.

Example:

Assume that the job container "myContainer" is still unsealed and containing a job "prod xy 99/aa:enable trace" that is no longer needed. The following command will remove this job from the container:

```
cadsmcmd jobContainer action=removeJob name="myContainer" jobName= job "prod xy
99/aa:enable trace"
```

renewJob—Renew Job

This command renews a failed job of the specified job container. The job options for the renewed job are modified according to the parameters and their coded values. The options belonging to parameters that are not coded remain unchanged.

This action has the following format:

```
jobcontainer action=renewJob

name=job_container_name
    jobName=job_name
    [newName=new_container_name]
    [newJobName=new_job_name]
    [deliveryTime="YYYY-MM-DD hh:mm"]
    [atTime="YYYY-MM-DD hh:mm"]
    [after={exacttime|boottime}]
    [calendarname=delivery_calendar]
    [preaction={none|reboot|logoff}]
    [postaction={none|reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}]
```

```
[promptUser[={y|n}]]
[allowCancel[={y|n}]]
[execTimedOut[={y|n}]]
[prompt=days.hours]
[offline[={y|n}]]
[runAtShutdown[={y|n}]]
[preventLogon[={y|n}]]
[globalTime[={y|n}]]
[nocalendar[={y|n}]]
[resolveQuery[={y|n}]]
[parameters=user_parameters]
[jobTimeout=d.h]
[userJobMessage]
[userMessage]
```

after

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

allowCancel

If "allowCancel" or "allowCancel=y" is coded, the user will be granted to cancel the job's execution.
If "allowCancel=n" is coded, the user will not be granted.

This parameter should only be coded, if "promptUser" is already enabled or will be enabled with this call, otherwise this setting will be ignored. No warning is given.

atTime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

calendarname

Specifies the name of the calendar that controls the time when the evaluation can take place.

deliveryname

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

execTimedOut

If "execTimedOut " or "execTimedOut=y" is coded, the job's execution will be automatically started when the user prompt times out.
If "execTimedOut=n", the job will not automatically be started.

This parameter should only be coded, if "promptUser" is already enabled or will be enabled with this call, otherwise this setting will be ignored. No warning is given.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

jobName

Specifies the name of the job to be renewed.

jobTimeout

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

name

Specifies the name of the job container of which a failed job is renewed.

newJobName

Specifies the new job name. If not coded the job name of the job to be renewed is used.

newName

Specifies the new name of the job container that will contain the renewed job. If this parameter is not coded a default container name of the form

```
<job name> [<timestamp>]
```

is generated.

noCalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

offline

If "offline" or "offline=y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.
If "offline=n" the connection will not be released.

parameters

Specifies the user parameters for the procedure.
Multiple user parameters must be separated by spaces inside double quotes.
If an empty string is coded, then no parameters are assumed.

postaction

Specifies the necessary actions that must take place after completion of the specified procedure. The following values are valid:

none

Performs no post-action.

reboot

Restarts the system after completion of the procedure.

logoff

Logs off the user after completion of the procedure.

rebootAtEnd

Restarts the system after completion of all the jobs of this container.

logoffAtEnd

Logs off the user after completion of all the jobs of this container.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

If the parameter is not coded, the default is given by the related procedure's job option value.

preaction

Specifies that necessary actions must take place before the specified procedure starts. The following values are valid:

none

Performs no pre-action.

reboot

Restarts the system before the start of the procedure.

logoff

Logs off the user before the start of the procedure.

If the parameter is not coded, the default is given by the related procedure's job option value.

preventLogon

If "preventLogon" or "preventLogon=y" is coded, then the user logon will be rejected during the job's runtime. If a user is already logged on, the execution of the job is delayed until the user logs off.
If "preventLogon=n" the user logon will not be rejected.

prompt=d.h

Specifies the time period in which the user will be prompted for the job's start.

The period's format is "*d.h*" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0). If the specified value falls below the valid minimum, the value is replaced by the minimum. If the specified value exceeds the valid maximum, then the value is replaced by the maximum. No warning is given.

This parameter should only be coded, if "promptUser" is already enabled or will be enabled with this call, otherwise this setting will be ignored. No warning is given.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

If the parameter is not coded, the default is given by the related procedure's job option value.

resolveQuery

Evaluate a possibly related query group before the job is generated.

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

userJobMessage

Specifies the custom administrator message for a renewed job in a job container.

userMessage

Specifies the custom administrator message for a job container.

Note: Only jobs of a job container of type "noLinkage" can be renewed as single jobs. For any other type of job container only renewal of the container is possible.

Example:

Assume the job "prod 3 1.0:inst" of container "myContainer" has failed, but the problem has been resolved now. Therefore the job should be renewed. The renewed job should meet the following requirements:

- The renewed job is named "Renew 0".
- The renew job is stored at the container "prod 3 1.0:inst".
- The user will be prompted before execution.
- The user has the permission to cancel the job.

The following command will renew this job:

```
cadsmcmd jobContainer action=renewJob name="myContainer" jobName="prod 3 1.0:inst"
newName="prod 3 1.0:inst" newJobname=renew promptUser allowCancel
```

renewTarget—Renew Failed Jobs of a Specific Target Computer

This action will renew all failed jobs of a job container addressing a specific target computer for this computer.

This action has the following format:

```
jobcontainer action=renewTarget name=job_container_name
    [newName=new_container_name]
    {computer=computer_name}
    [userMessage]
```

name

Specifies the name of the job container of which failed jobs will be renewed.

newName

Specifies the new name of the job container for the renewed jobs. If this parameter is not coded the name of the original job container is used.

computer

Specifies the name of the computer of which the renewable jobs of the job container in question will be renewed.

The parameter can be coded more than once to address a list of targets.

userMessage

Specifies the custom administrator message for the renewed job container.

Note: If the job container is of type "synchronized job execution" then individual targets cannot be renewed, but only whole job containers.

Example:

Assume that problems on the system "KKKKK01B" have made the container "myContainer" fail, but the problems have been resolved now. Therefore, the failed jobs of the container "myContainer" for this target should be renewed. The renewed job should meet the following requirement:

- The renew job is stored at the container "renew KKKKK01B".

The following command performs this renew:

```
cadsmcmd jobContainer action=renewTarget name=myContainer newName="renew KKKKK01B"
computer=KKKKK01B
```

repositionJobs—Reposition Jobs in Job Container

This action allows you to reorder a job container by assigning a new order number to a job of the job container.

This action has the following format:

```
jobContainer action=repositionJobs
    name=job_container_name
    cjobname=job_name position=newposition
```

name

Specifies the name of the job container to be reordered.

jobName

Specifies the name of the job to be repositioned.

position

Specifies the new position of the job.

The position has to be greater than zero and should not exceed the number of jobs in the job container.

Note: The job container has to be unsealed.

Example:

Assume the job "prod 3 1.0:inst" is the second job of the unsealed job container "myContainer", but for integrity reasons it should be the first. The following command repositions the job:

```
cadsmcmd jobContainer action=repositionJobs name=myContainer jobName="prod 3
1.0:inst" position=1
```

showAttrJobs—Show Job Attributes

This action lists the attributes of a job from the specified job container.

This action has the following format:

```
jobcontainer action=showAttrJobs
    name=job_container_name
    jobName=job_name
```

name

Specifies the name of the job container from where the job is taken.

jobName

Specifies the name of the job to be listed.

Example:

To see the attributes associated with job "prod 3 1.0:inst" of "myContainer" enter the following command:

```
cadsmcmd jobContainer action=showAttrJobs name="myContainer" jobName="prod 3
1.0:inst"
```

The output will be as follows:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "wwwww01b" as user "<default user>" ...OK.

------------------------------------------------------------------------------
List of attributes of job “prod 3 1.0:inst “ from container "myContainer"
------------------------------------------------------------------------------
Job name.............................: prod 3 1.0:inst
Software.............................: prod 3 1.0
Procedure............................: instCrash
Deliver at...........................: 20035-04-12 10:49
Activate at..........................: 2005-04-12 10:49
Status message.......................: OK
Overall state........................: ERROR
.....................................0 - job targets waiting
.....................................0 - job targets active
.....................................1 - job targets ok
.....................................1 - job targets failed
Job task.............................: install
Pre action...........................: no action
Post action..........................: no action
Prompt user..........................: Yes
Characteristic flags.................:
.....................................local time/exact
.....................................calendar controlled
Delivery calendar....................:
User parameters......................:
Prompt during........................: 1 Days, 0 Hours
Job timeout..........................: 7 Days, 0 Hours

sdcmd<A000000>: OK
```

showJobDetails—Show Job Details

This action shows job details of an evaluated job container.

This action has the following format:

```
jobcontainer action=showjobdetails
    name=job_container_name
    jobname=job_name
    destination=destination_name
    [task={install|uninstall|activate|configure|deliver|undeliver }]
```

name

Specifies the name of the job container.

jobname

Specifies the name of the job to be evaluated.

The job has to be a member of the specified job container.

destination

Specifies the destination addressed by the job for which the evaluation should take place.

task

Specifies the task of the job to be evaluated.

It is one of the following:

install

installation procedure

uninstall

uninstallation procedure

activate

activation procedure

configure

configuration procedure

deliver

A delivery procedure

undeliver

An un-delivery procedure.

Example:

To retrieve detailed information about the job "prod 3 1.0:inst" related to the target KKKKK01B, enter the following command:

```
cadsmcmd jobContainer action=showJobDetails name="myContainer" jobName="prod 3
1.0:inst" destination=KKKKK01B


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM


------------------------------------------------------------------------------
Details for job "prod 3 1.0:inst" to
........."KKKKK01B" of "myContainer"
------------------------------------------------------------------------------
Comment.............................:
Job container status..................: failed
Job container properties..............: 9
.....................................No linkage
.....................................Rollback previous jobs on failure
Sealed..............................: Yes
Creation date.......................: 2005-04-12
Creation time.......................: 08:30
Change date.........................: 2005-04-12
Change time.........................: 12:16
Number of contained objects.........: 4
```

```
Job name.............................: prod 3 1.0:inst
Overall state........................: ERROR
.....................................0 - waiting
.....................................0 - active
.....................................1 - ok
.....................................1 - failed
..Job order number...................: 1
..Creation date......................: 2005-04-12
..Creation time......................: 10:57
..Delivery date......................: 2003-04-12
..Delivery time......................: 10:49
..Activation date....................: 2005-04-12
..Activation time....................: 10:49
..Delivery calendar..................:
..Timeout <days/hours>...............: 7 / 0
..Job startup........................: local time/exact
..Job permission.....................: 7
.....................................renew allowed
.....................................cancel allowed
.....................................recover allowed
..Pre action.........................: no action
..Post action........................: no action
..Job operation mask.................: 134
.....................................prompt user
.....................................calendar controlled
.....................................Repeat for every evaluation
..User parameters....................:
..Job execution mode.................: 1
.....................................execute
..Prompt timeout <days/hours>........: 1 / 0
..Job task...........................: 1 - install
..Targetsystem.......................: KKKKK01B
....Task.............................: 1
```

```
.....................................install
....Procedure name.....................: inst
....Ordered by.........................: KKKKK01
....Job state..........................: EXECUTION_ERROR
....Error message......................: Exit code 2 indicates possible error
....Error cause........................: 228001
....Comment............................:
....Computer installation procedure....: inst
....Activation date....................: 2005-04-12
....Activation time....................: 10:49
....Creation date......................: 2005-04-12
....Creation time......................: 12:16
....Completion date....................: 2005-04-12
....Completion time....................: 12:16
....Data sent in per cent..............: 100

SDCMD<A000000>: OK
```

Note: If the job presented is a reinstallation job, there will be a corresponding entry in the output (after "Job operation mask"), otherwise the line is missing.

managedcomputer—Manage Computer Configurations

Use the managedcomputer command to manage computer configurations.

This command is supported on domain managers only.

This section contains the following topics:

listConfigs—List Computer Configurations

The action lists the available configurations associated with the specified computers. Reported configurations are not listed with this action.

This action has the following format:

```
managedcomputer action=listConfigs
    name=computer_name
    [status=configuration_status]
    [DETAILED [DESC]]
```

name

Specifies the name of the computer of which the associated configurations will be listed.

The name might contain wildcards.

In this case for all those computers matching the pattern by their names the configurations will be listed.

status

Specifies the status of the configurations.

DETAILED

Information is listed in detail

Otherwise only "Identifier", "Target", "Status", and "Activation Date" are listed with the configuration.

DESC

If coded, the parameter description information is listed,otherwise not.

Example:

```
cadsmcmd managedcomputer action=listconfigs name=myComp DETAILED
```

The following is the output:

```
--------------------------------------------------------------------------------
List Configurations of myComp[DETAILED]
--------------------------------------------------------------------------------
Computer name.......................: myComp
Configuration Name..................: myComp_update:1132585401
..Identifier........................: F492710E-5A9F-11DA-89FD-AB3891E877B6
..Target............................: myComp
....Type............................: COMPUTER
....Identifier......................: 7EF10641-43DC-4DFA-82B2-9D595F58081B
..State.............................: active
..Activation Date...................: 2006-11-21 16:03
..Tailored Settings
..Policies
```

```
Configuration Name.....................: myComp: API [2005/11/21 16:02:20 UTC]
..Identifier...........................: 31D27494-5AA8-11DA-89FD-AB3891E877B6
..Target...............................: myComp
....Type...............................: COMPUTER
....Identifier.........................: 7EF10641-43DC-4DFA-82B2-9D595F58081B
..State................................: scheduled
..Tailored Settings
..Policies
....Policy Name........................: myPol
....Type...............................: COMPUTER
....Comment............................:
....Status.............................: sealed
....Parameter Sections
......Parameter Section Name...........: /itrm/usd/Server
........Identifier.....................: 09625218-5AA8-11DA-89FD-AB3891E877B6
........Parameters
..........Parameter Name...............: WaitBetweenStageChecks
............Identifier.................: 0977C742-5AA8-11DA-89FD-AB3891E877B6
............Value......................: 10
```

reportCurrentConfig—Report Current Computer Configuration

This action lists the reported current configuration of the specified computers.

This action has the following format:

```
managedcomputer action=reportCurrentConfig
    name=computer_name
    [DETAILED [DESC]]
```

name

Specifies the name of the computer of which the current configuration report is to be listed. The computer name might contain wildcards. In this case for all those computers matching the pattern by their names the report will be listed.

DETAILED

Information is listed in detail Otherwise "Status" and "Request Date" or "At" are listed for the specified computer.

DESC

If coded, the parameter description information is listed,otherwise not.

Example:

```
cadsmcmd managedcomputer action=reportcurrentconfig name=mycomp
```

The following is the output:

```
------------------------------------------------------------------------------
List Current Configuration Reports for myComp
------------------------------------------------------------------------------
Computer name.........................: myComp
..Status..............................: received
..At..................................: 2006-11-21 15:51

Number of Configurations: 0
Number of reports listed: 1
```

requestConfigReport—Request a Configuration Report

The action requests a configuration report for a specified computer.

This action has the following format:

```
managedcomputer action=requestConfigReport
    name=computer_name
    [WAIT]
```

name

Specifies name of the computer to report its current configuration.

The name might contain wild cards. In this case the request is launched for all those objects where the name matches the specified pattern.

WAIT

If coded the action will wait for the completion of the order, otherwise it just returns after the order has been launched.

manager—Manage DSM Managers

This section contains the following topics:

list—List Managers

Use this action to list the attributes of the manager.

This action has the following format:

```
manager action=list
```

Output format:

The information provided is formatted as follows:

```
------------------------------------------------------------------------
Attributes of manager "«manager name»"
------------------------------------------------------------------------
Manager name..:......«name of the manager»
..Manager type......:.......{domain|enterprise}
..Manager address...:.......«manager address»
..Description : «description of manager»
..Domain database name.....:«domain database label»
       •
       •
       •
Manager name..:......«name of the manager»
..Manager type......:.......{domain|enterprise}
..Manager address...:.......«manager address»
..Description : «description of manager»
..Domain database name.....:«domain database label»
```

modify—Modify Manager Atributes

Use this action to modify attributes of the manager.

The action has the following syntax:

```
manager action=modify
    [name=manager_name]
    [newName=new_manager_name]
    [comment=description_for_manager]
```

name

Specifies the name of the manager whose attributes will be modified.

If not coded then the manager is taken with which the CADSMCMD currently is in session.

newName

specifies a new name for the manager.

comment

specifies a new description for the manager.

Note: The manager has to be located in the domain on that the CADSMCMD currently works.

showAttr—Show Manager Atributes

Use this action to list the attributes of the manager.

The action has the following syntax:

```
manager action=showAttr
    [name=manager_name
    [domain=domain_name]]
```

name

Specifies the name of the manager of which the attributes will be listed.

If not coded then the manager is taken with which the CADSMCMD is in session.

domain

Name of the domain the manager belongs to.

Output format:

The information listed is formatted as follows:

```
------------------------------------------------------------------------
Attributes of manager "«manager name»"
------------------------------------------------------------------------
Manager name..:.....«name of the manager»
..Manager type.....:......{domain|enterprise}
..Manager address...:......«manager address»
..Description...:......«description»
..Created at..:.....«yyyy-mm-dd HH:MM:SS»
..Last modified at..:......«yyyy-mm-dd HH:MM:SS»
..Domain database name.....:......«database name»
....Network address.:......«database host name»
....Instance..:.....«database instance»
....Database type...:......{MS SQL|Ingres|Oracle}
..Components
....Name.....:......«component name»
......Version:......«component version»
       •
.......•
.......•
....Name.....:......«component name»
......Version:......«component version»
[..Enterprise database name.:.....«enterprise database label»
....Enterprise database address...:.....«enterprise database address»]
[..Sub domain database name:......«sub domain database label»
....Sub domain database address...:......«sub domain database address»
....•
....•
....•
..Sub domain database name.:......«sub domain database label»
....Sub domain database address...:......«sub domain database address»]
```

progress—Checking Distribution Progress

The progress command is used to return the status about a distribution initiated by one of the software distribution commands, or initiated by another SD user.

The progress command is valid only when issued for an enterprise manager.

This command has the following format:

`progress name=`*`job_container_name`*

name

Specifies the name of the job container against which to perform a status check.

rdereg—Deregistering Items Remotely

Use the rdereg command to send an order to deregister (remove) software items from the software library, at one or more domain managers. The order originates from the enterprise manager. The deregistration order is sent from the enterprise manager to the specified CA ITCM domains or domain groups.

This command is valid for enterprise managers only.

It has the following format:

```
rdereg  item=item_name
    version=version_name
    {{area=area_name} | {domain=domain_name} | toAllAreas}
    [cname=name]
    [sendTime="YYYY-MM-DD hh:mm"]
    [haltTime="YYYY-MM-DD hh:mm"]
```

area

Specifies the name of a CA ITCM domain where the item is to be deregistered.

The item can specified more than once to address a list of CA ITCM domains.

The item should not be coded with the parameters domain or toAllAreas.

cname

Specifies the the optional unique name of a job container.

If a job container / distribution with the specified name already exists an error will be reported.
If "cname" is not coded, a generic name is generated.

domain

Specifies the name of a CA ITCM domain group where the item is to be deregistered.

The parameter can be specified more than once to address a list of CA ITCM domain groups.

The parameter should not be specified with the area or toAllAreas parameters.

haltTime

Specifies the date and time at which a distribution order should be halted. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm".

item

Specifies the name of the item to be deregistered or removed from one or more local libraries.

sendTime

Specifies the time to send the associated distribution to the addressed domain managers. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm". If the parameter is not coded, the actual date and time is used.

toAllAreas

If coded the related distribution will be sent to all CA ITCM domains registered at the enterprise manager.

version

Specifies the version of the item to be deregistered or removed.

Example:

```
cadsmcmd rdereg item="Software TestC" version=0.46
area=N_LOCAL_SRV area=N_LOCAL_SRV_02 area=WO_LOCAL_SRV cname=TestCont4
sendtime="2005-03-05 18:35" halttime="2005-03-04 23:30"
```

rderegproc—Deregistering Procedures Remotely

Use this command to send an order to deregister (remove from the library) an item procedure associated with a software program in the library at one or more domain managers. The item procedure cannot be an embedded item procedure.

The command is valid for enterprise manager only.

This command has the following format:

```
rderegproc  item=item_name
    version=version_name
    procedure=procedure_name
    {{area=area_name} | {domain=domain_name}| toAllAreas}
    [cname=name]
    [sendTime="YYYY-MM-DD hh:mm"]
    [haltTime="YYYY-MM-DD hh:mm"]
```

area

Specifies the name of a CA ITCM domain where the procedure is to be deregistered.

The item can specified more than once to address a list of CA ITCM domains.

The item should not be coded with the parameters domain or toAllAreas.

cname

Specifies the the optional unique name of a job container from where the jobs to be recovered will be taken.

If a job container / distribution with the specified name already exists an error will be reported.
If "cname" is not coded, a generic name is generated.

domain

Specifies the name of a CA ITCM domain group where the procedure is to be deregistered.

The parameter can be specified more than once to address a list of CA ITCM domain groups.

The parameter should not be specified with the area or toAllAreas parameters.

haltTime

Specifies the date and time at which a distribution order should be halted. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm".

item

Specifies the name of the item being deregistered

procedure

Specifies the name of a procedure.

sendTime

Specifies the time to send the associated distribution to the addressed domain managers. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm". If the parameter is not coded, the actual date and time is used.

toAllAreas

If coded the related distribution will be sent to all CA ITCM domains registered at the enterprise manager.

version

Specifies the version of the item being deregistered.

Example:

```
cadsmcmd rderegproc item="Software TestC" version=0.46 procedure=Install2
area=N_LOCAL_SRV area=N_LOCAL_SRV_02 area=WO_LOCAL_SRV cname=TestCont3
sendtime="2005-03-06 18:45" halttime="2005-03-06 23:30"
```

recover—Recovering Failed Installations

The recover command recovers the failed installations. The command also cleans up the failed installations.

The recover command is valid only on the domain managers.

This command has the two following formats:

```
recover {item=item_name version=version_name
    installedwith=install_procedure
    procedure=un_install_procedure_name
    [{computer=computer_name}]
    [after={exacttime|boottime}]
    [nocalendar[={y|n}]
    [offline[={y|n}]]
    [deliverytime="YYYY-MM-DD hh:mm" ]
    [attime="YYYY-MM-DD hh:mm" ]
    [promptUser[={y|n}]]
    [bootbefore[={y|n}]
    [bootafte[={y|n}]
    [timeout=hours]
    [calendarname=delivery_exclusion_calendar]
    [runAtShutdown[={y|n}]
    [userJobMessage]
    [userMessage]

recover cname=name
    [item=item_name
    version=version_name
    installedwith=install_procedure
    procedure=un_install_procedure_name
    [after={exacttime|boottime}]
    [globalTime[={y|n}]]
    [nocalendar[={y|n}]
    [offline[={y|n}]]
    [deliverytime="YYYY-MM-DD hh:mm" ]
    [attime="YYYY-MM-DD hh:mm" ]
    [promptUser[={y|n}]]
    [bootbefore[={y|n}]
    [bootafte[={y|n}]
    [timeout=hours]
    [calendarname=delivery_exclusion_calendar]
    [nopurge]
    [runAtShutdown[={y|n}]
    [userJobMessage]
    [userMessage]
```

after={exacttime | boottime }

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

atTime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

bootafter

Indicates that the target computer is to be rebooted before the procedure starts.

bootbefore

Indicates that the target computer has to be rebooted after the procedure has completed.

calendarname

Specifies the name of a calendar, which specifies the periods where deliveries are excluded.

cname

Specifies the name of the job container from where the jobs to be recovered will be taken.

computer

Specifies the name of the computer.

deliveryTime

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time will be used.

globaltime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

installedWith

Specifies the name of the install procedure.

This parameter should be coded together with the parameter item only.

item

Specifies the name of the item being recovered.

This parameter should be coded with the parameters version, installedWith and procedure only.

If the parameter item is not coded, then all recoverable jobs of the specified container will be recovered using the first uninstall procedure related to an item to be recovered.

noCalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

nopurge

The job container coded in cname is kept if the parameter is coded, otherwise the container will be deleted.

offline

If "offline" or "offline=y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.
If "offline=n" the connection will not be released.

If the parameter is not coded, the default is given by the related procedure's job option value.

procedure

Specifies the name of the "uninstall" procedure to be performed.

This parameter should be coded with the parameter item only.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

If the parameter is not coded, the default is given by the related procedure's job option value.

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

If the parameter is not coded, the default is given by the related procedure's job option value.

timeout

Specifies the timeout specifies the expiration period of a job in the form “h” where h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
timeout should not be coded with the jobTimeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 168

version

Specifies the version of the item being recovered.

This parameter should be coded with the parameter item only.

userJobMessage

Specifies the custom administrator message for a failed job in a job container.

userMessage

Specifies the custom administrator message for a job container.

Note: If the cname variant of the command is coded then the block with the parameters *item version installed* is optional. If these parameters are not coded, all recoverable jobs of the container will be recovered and the uninstall procedure used is the first one in the list of the uninstall procedures found which are assigned to the related products.

Example:

The following example illustrates how the recover command can be used to recover a failed installation, from the domain manager, on computer N_LOCAL_SRV:

```
cadsmcmd recover item="Software TestC" version=0.49A
installedwith=Install2 procedure=UninstallC
computer=N_LOCAL_SRV after=exacttime
attime="2005-10-10 10:55" promptuser timeout=168
```

regproc--Registering Procedures

Use the regproc command to register an embedded or additional item procedure with an existing software program. The item procedure must be in a directory accessible to the enterprise or domain manager.

The syntax of the command is as follows:

```
cadsmcmd regproc        item=item_name
    version=version_name
    task={install | activate | configure | uninstall}
    procedure=procedure_name
    os=os
```

```
type={executable | command | SWD | MSI | SXP | PKG | PIF | IPS | PALM | WINCE|
RPM | NOKIA}
[enduser]
[notRAC]]
[externalproc]
file=file_name
path=path_name
[parameters=procedure_parameters]
[defaultProc][query=query_name][sep=separator_sign]
[comment=comment]
[usedefaultMsg]
```

comment

Specifies a comment for the procedure.

defaultProc

If coded, the procedure registering becomes the default procedure of all procedures of the same type.

enduser

(Optional) Indicates if the procedure can be scheduled for execution by an end user through the Catalog GUI.

externalproc

(Optional) Indicates that the procedure given in the procedure parameter is external (the procedure has to be provided by the target systems in question).

file

Specifies the name of the procedure file.

item

Specifies the name of the item to which this procedure is being associated as an added item procedure.

notRAC

(Optional) Specifies that the procedure is not to be taken into account at Reinstall After Crash (RAC).

os

Specifies the operating system type that the procedure will support.

Values:

```
os={AIX|AIX_3.2_RS/6000|AIX_4.1_RS/6000
...|AIX_4.2_RS/6000|AIX_4.3_RS/6000
...|AIX_5.0_RS/6000|AIX_5.1_RS/6000
...|AIX_5.2_RS/6000|AIX_5.3_RS/6000
...|AIX_5.4_RS/6000|AIX_5.5_RS/6000
...|AIX_5.6_RS/6000|AIX_5.7_RS/6000
...|AIX_5.8_RS/6000|ANY|COMPAQ_TRU64_(DEC)
...|COMPAQ_TRU64_5.X_ALPHA
...|DEC_3.0_ALPHA|DEC_3.2_ALPHA
...|DEC_4.0_ALPHA|DGUX|DGUX_4.1_INTEL
...|DGUX_4.1_MC88|DGUX_4.2_INTEL
...|DOS|DRS/NX_7MP_SPARC|DRS/NX_7_SPARC
...|DRS/NX_SPARC|DYNIX|DYNIX_4.2.X_INTEL
...|DYNIX_4.4.X_INTEL|HPUX
...|HPUX_10|HPUX_10_700|HPUX_10_800
...|HPUX_11|HPUX_11I_V2|HPUX_11_700
...|HPUX_11_800|HPUX_11_IA64
...|HPUX_12|HPUX_12I_V2|HPUX_12_700
...|HPUX_12_800|HPUX_12_IA64
...|HPUX_9|HPUX_9_700|HPUX_9_800
...|IRIX|IRIX_6.2|IRIX_6.3
...|IRIX_6.4|IRIX_6.5|LINUX
...|LINUX_2.1_INTEL|LINUX_2.2_INTEL
...|LINUX_2.2_S/390|LINUX_2.3_INTEL
...|LINUX_2.3_S/390|LINUX_2.4_INTEL
...|LINUX_2.4_S/390|LINUX_2.5_INTEL
...|LINUX_2.5_S/390|LINUX_2.6_INTEL
...|LINUX_2.6_INTEL_64-BIT
...|LINUX_2.6_S/390|MACINTOSH
...|MACINTOSH_10.0_PPC|MACINTOSH_10.1_PPC
...|MACINTOSH_10.2_PPC|MACINTOSH_10.3_PPC
...|MACINTOSH_10.4_PPC|NCR_SST
...|NCR_SST_S4I_INTEL|NCR_SST_S4_INTEL
...|NCR_SV|NCR_SV_2X_INTEL
...|NCR_SV_3X_INTEL|NETWARE
...|NETWARE_3_INTEL|NETWARE_4_INTEL
...|NETWARE_5_INTEL|NETWARE_6_INTEL
...|NOKIA_ADMINSUITE|NOKIA_ADMINSUITE_1.0
...|OPENVMS|OPENVMS_5.5_VAX
...|OPENVMS_6.0_ALPHA|OPENVMS_6.0_VAX
...|OPENVMS_6.1_ALPHA|OPENVMS_6.1_VAX
...|OPENVMS_6.2_ALPHA|OPENVMS_6.2_VAX
...|OPENVMS_7.0_ALPHA|OPENVMS_7.0_VAX
...|OPENVMS_7.1_ALPHA|OPENVMS_7.1_VAX
...|OPENVMS_7.2_ALPHA|OPENVMS_7.2_VAX
```

```
...|OS/2_16-BIT|OS/2_32-BIT
...|PALM_OS|SCO|SCO_3.2_V5_INTEL
...|SCO_5.2_V5_INTEL|SINIX
...|SINIX_5.42_R4000|SINIX_5.43_R4000
...|SINIX_5.44_R4000|SINIX_5.45_R4000
...|SOLARIS|SOLARIS_10_INTEL
...|SOLARIS_10_SPARC|SOLARIS_2.3_SPARC
...|SOLARIS_2.4_INTEL|SOLARIS_2.4_SPARC
...|SOLARIS_2.5_INTEL|SOLARIS_2.5_SPARC
...|SOLARIS_2.6_INTEL|SOLARIS_2.6_SPARC
...|SOLARIS_7_INTEL|SOLARIS_7_SPARC
...|SOLARIS_8_INTEL|SOLARIS_8_SPARC
...|SOLARIS_9_INTEL|SOLARIS_9_SPARC
...|SUNOS|SUNOS_4.1.4_SPARC
...|UNIX|UNIXWARE|UNIXWARE2.0_INTEL
...|UNIXWARE2.1_INTEL|UNIXWARE7.X_INTEL
...|WINDOWS_16-BIT|WINDOWS_32-BIT
...|WINDOWS_95|WINDOWS_98
...|WINDOWS_ME|WIN_2000|WIN_2000_INTEL
...|WIN_CE|WIN_CE_2.00_MIPS
...|WIN_CE_2.00_SH3|WIN_CE_2.01_MIPS
...|WIN_CE_2.01_SH3|WIN_CE_2.11_ARM
...|WIN_CE_2.11_MIPS|WIN_CE_2.11_SH3
...|WIN_CE_2.11_SH4|WIN_CE_3.00_ARM
...|WIN_CE_3.00_MIPS|WIN_CE_3.00_PPC2002_ARM
...|WIN_CE_3.00_SH3|WIN_CE_3.00_SH4
...|WIN_CE_4.20_MOBILE2003_ARM
...|WIN_LONGHORN_SERVER_INTEL
...|WIN_LONGHORN_SERVER_ITANIUM
...|WIN_LONGHORN_SERVER_X64
...|WIN_NT|WIN_NT_ALPHA|WIN_NT_INTEL
...|WIN_NT_MIPS|WIN_SERVER_2003
...|WIN_SERVER_2003_INTEL
...|WIN_SERVER_2003_ITANIUM
...|WIN_SERVER_2003_X64|WIN_VISTA_INTEL
...|WIN_VISTA_ITANIUM|WIN_VISTA_X64
...|WIN_XP|WIN_XP_INTEL|WIN_XP_ITANIUM

...|WIN_XP_X64}
```

Note: When you issue the regproc command in verbose mode, a list of valid values for your actual version is displayed.

parameters

Parameters to be passed to the procedure when executed.

path

Indicates one of these three cases, depending on usage:

internal procedure

The procedure is an integral part of the product package. Therefore, the path is expected to be the relative path from the product root to the procedure file.

If, for example, the product has been loaded form c:\swtest1 (the product root) and the procedures are located at c:\swtest1\procs then code path=PROCS to locate the procedure.
If the procedure lies also at c:\swtest1 then code path=\ on Windows and path=./ on Linux.
(See also: Backslash (Windows) and Forward Slash (Linux))

added procedure

The procedure is not an integral part of the package but available somewhere on the manager. Therefore, the path is to be the absolute access path to the procedure file on the manager.

If, for example, the product has been loaded from c:\swtest1 but the procedure to be assigned is located at d:\procpool on the manager where the command is launched, then code path=d:\procpool.

external procedure

The procedure neither is an integral part of the package nor is it available on the manager where the command is launched, but it should be available at all those target computers where it should later run on.

Then the procedure has to be located at the same location on all those target computers to be addressed.

For example, provide the procedure at c:\usd_supp\procs at the target computers in question. Then code path=c:\usd_supp\procs.

Note: This parameter is not mandatory if the externalProc argument is coded. In this case the information passed with the file argument will be used to determine the procedure to be invoked at the target systems.
If the the externalProc argument is not coded then this parameter remains mandatory.

procedure

Specifies the name of a procedure. It will be registered.

query

Specifies the name of an existing query that describes the prerequisites for running this procedure. If not coded then no prerequisites are set.

The query name syntax is of the following syntax:

{*query_name* | (*query_name separator_sign domain_name*)}

The domain_name specifies the name of the domain where this query has been created. The separator_sign separates the query_name of the domain_name. The separator_sign is defined with the sep parameter. If only the query_name is specified, then the membership of the query to a domain is not checked.

sep

Specifies the separator sign used with this action. The default separator sign is the dot ".".

task

Specifies the type of task to be performed by the item procedure.

Install

Registers an installation procedure

Activate

Registers an activation procedure

Configure

Registers a configuration procedure

Uninstall

Registers a uninstallation procedure

type

Specifies the procedure type.

Valid values are:

executable

executable file (.exe or .com)

command

command file (.cmd or .bat)

swd

Software Detector script

msi

Microsoft Windows Installer package file

sxp

SD packaging format for Windows

pkg

SD packaging format for UNIX

pif

Product Interchange File: SD packaging format for UNIX

rpm

Linux RedHat Package Manager packaging format

ips

<adms> script file with the extension .ips

palm

PALM application file with the extension .prc

wince

Windows CE cabinet file with the extension .cab

nokia

NOKIA 9210 Communicator file

usedefaultMsg

(Optional) If the usedefaultMsg argument is coded, the comment associated with the item procedure is used as the user job message.

version

Specifies the version of the item.

Example:

In the following example, a procedure is registered from a Windows server command line, and made available for request through the Catalog. Since the procedure file, install1.cmd, does not reside in the current directory or in the system path, but on the target system, it is an external procedure and the full path is specified.

```
cadsmcmd regproc item="Software Test" version=0.45 task=install
procedure=Install os=WINDOWS_32-BIT type=command enduser externalproc
file=install.cmd path=c:\test\swtest1 comment="Install Proc 1"
```

In the following example the argument externalProc is coded. The path parameter is not used. The executable associated with the procedure conf_ext is expected to be the program configure.exe which is to be found at c:\myProcs\Prod.

```
regproc item=prod1 version=1.2/98 task=configure procedure=conf_ext os=WIN_NT
type=executable externalProc file=c:\myProcs\Prod1
```

Also in the next example the path parameter is not used. The executable associated with the procedure act_ext is the program act.exe to be found within the %PATH% of the target systems.

```
regproc item=prod1 version=1.2/98 task=activate procedure=act_ext os=WIN_NT
type=executable externalProc file=c:\myProcs\Prod1\act.exe
```

In the following example the path parameter is used: the executable associated with the procedure inst_ext is expected to be the program install.exe and it is to be found at c:\myProcs\Prod1 on the target systems.

```
regproc item=prod1 version=1.2/98 task=install procedure=inst_ext os=WIN_NT
type=executable externalProc file=install.exe path=c:\myProcs\Prod1
```

regsw--Registering Software

The regsw command lets you:

- Register a new software program
- Register a new program version, based on a program version already registered in the library

This command has the following format:

```
regsw   {item=item_name
    version=item_version
    [basedonitem=baseproduct basedonversion=baseversion] path=sourcepath
    [comment=comment]
    [supplier=supplier]
    [procedures=procedurelistname]}
    [nochecksum]
```

basedonitem

Specifies the product name of the original item on which this new item is based.

basedonversion

Specifies the version of the base product.

comment

Specifies a comment about this item version.

item

Specifies the name of the new item.

noChecksum

If this optional parameter is coded, there is no integrity checking by checksum.

By default, if a package is defined by "regsw", the checksum control is enabled. If the checksum control must be disabled, this parameter must be coded.

In verbose mode, you are asked to enter "Yes" or "No" for the "NoChecksum" parameter. Enter "Yes" to set the parameter, "No" to disable the parameter.

The verbose default for this parameter is "No", that is, the checksum control is enabled.

path

Specifies the directory on the manager that holds the image of the item being registered.

If "*sourcepath*" refers to a mapped drive, this drive must be accessible to the SD service.

procedures

This parameter specifies the name of a file that defines the embedded procedures of the item to be registered.

SD must have access to this file.

The procedures are defined by regproc commands.

For the contents of this file the following holds:

- Regrpoc can be coded without item and version parameters. They are optional for such a file. But if they are coded, they have to refer to the product being registered, otherwise an error is reported.
- Regproc can span over serveral lines.
- Comments can be added. The have to be proceded by a hash sign (#).
- Parameter values containing blanks have to be enclosed in double quotes (''),

If the procedures parameter is coded, the product is automatically sealed after a successful registration, otherwise it remains unsealed for further processing.

supplier

Specifies the supplier of the item.

version

Specifies the version of the item.

The following example shows how the regsw command would be issued from a Linux server to register version 0.45 of a software package called "Software Test".

```
cadsmcmd regsw item="Software Test" version=0.45 path=/usr/test/swtest1
comment="Test software registration" supplier=CA procedures=/usr/test/proclist.txt
```

The following example illustrates how to register the same software package from a **Windows** server, and includes four embedded item procedures. The procedures parameter provides the name of the file where the procedures are listed.

```
cadsmcmd regsw item="Software Test" version=0.45 path=c:\test\swtest1
comment="Test software registration"supplier=CA procedures=c:\test\proclist.txt
```

Following is a sample file containing a list of procedures:

This file contains a list of all embedded item procedures for Software Test, Version 0.45

```
regproc task=install procedure="Install Proc 1" os=WINDOWS_32-BIT type=command
enduser file=install1.cmd path=\

regproc task=activate procedure="Activate Proc 1" os=WINDOWS_32-BIT type=command
enduser file=activ1.cmd path=\

regproc task=configure procedure="Configure Proc 1" os=WINDOWS_32-BIT type=command
enduser file=config1.cmd path=\

regproc task=uninstall procedure="Uninstall Proc 1" os=WINDOWS_32-BIT type=command
enduser file=uninst1.cmd path=\
```

The path is "\", because the procedures reside in the same directory as the software itself. If the procedures resided in the subdirectory "C:\test\swtest1\proc", then "path=\proc" should have been used in the procedures file. Path is defined in the same manner on Linux servers, with the obvious change of directory separator to "/".

renew—Renewing a Distribution

The renew command is valid only when issued from the enterprise manager.

Renew is used to renew a failed or halted distribution.

This command has the following format:

```
renew name=distribution_name
    [sendTime="YYYY-MM-DD hh:mm"]
    [haltTime="YYYY-MM-DD hh:mm"]
```

halttime

Specifies the date and time at which a distribution order should be halted. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm".

name

Specifies the name of the distribution to be renewed.

sendtime

Specifies the time to send the associated distribution to the addressed domain managers. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm". If the parameter is not coded, the actual date and time is used.

Example:

Consider the following example, issuing the cadsmcmd progress command:

```
cadsmcmd progress name=TestCont11
```

This command shows TestCont11 as a halted distribution:

```
Distribution name: TestCont11, Status Halted.
SendDateTime: 2003/03/05 18:04, InstallDateTime: 2003/03/05 18:10, HaltDateTime:
2003/03/05 18:15
Nr of Areas Total: 2, Waiting: 0, Active; 0, in Error: 1, OK: 1.

Area: N_LOCAL_SRV at Downtown 100 % sent. Status OK.
Area: N_LOCAL_SRV_02 at Rural Area 0 % sent. Status Halted.
SDCMD<A000000> : OK
```

The cadsmcmd renew command is issued to renew the halted TestCont11 distribution:

```
cadsmcmd renew name=TestCont11
```

Reissuing the cadsmcmd progress command shows that the TestCont11 distribution has been renewed and is in active status.

```
Distribution name: TestCont11, Status Active.
SendDateTime: 2003/03/05 18:04, InstallDateTime: 2003/03/05 19:10, HaltDateTime:
2003/03/05 22:30
Nr of Areas Total: 2, Waiting: 0, Active; 1, in Error: 0, OK: 1.

Area: N_LOCAL_SRV at Downtown 100 % sent. Status OK.
Area: N_LOCAL_SRV_02 at Rural Area 0 % sent. Status Active.
SDCMD<A000000> : OK
```

renewjob—Renewing a Job

The renewjob command renews a failed job. This command is valid only when issued from the domain manager.

This command has the following format:

```
renewjob
    item=item_name
    version=version_name
    procedure=procedure_name
    installedwith=install_procedure_name
    task={install | activate | configure | uninstall}
    [deliverytime="yyyy-mm-dd hh:mm" ]
    [calendarname=delivery_exclusion_calendar]
    [attime="yyyy-mm-dd hh:mm" ]
    [timeout=hours]
    [runAtShutdown[={y|n}]]
    [userJobMessage]
    [userMessage]
```

or

```
renewjob
    cname=[name]
    [item=item_name
    version=version_name
    procedure=procedure_name ]
    [deliverytime="yyyy-mm-dd hh:mm" ]
    [attime="yyyy-mm-dd hh:mm" ]
    [calendarname=delivery_exclusion_calendar]
    [timeout=hours]
    [runAtShutdown[={y|n}]]
    [userJobMessage]
    [userMessage]
```

atTime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

calendarname

Specifies the name of a calendar, which specifies the periods where deliveries are excluded

cname

Specifies the optional unique name of a job container that contains the jobs to be renewed.

The parameters item, version, and procedure are optional when cname is coded:

If the latter ones are coded, only the jobs of the container, identified by these parameters, are renewed.

If they are not coded, then all renewable jobs of the specified container are renewed.

If cname is not coded, then the job to be renewed is the first renewable job found that meets the specification of the parameters item, version, installedWith, procedure, and task.

deliverytime

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

installedWith

Specifies the name of the install procedure.

The job to be renewed has been installed with this procedure.

The parameter is ignored, if cname is coded.

item

Specifies the name of the item. The job to be renewed has to deal with this item.

procedure

Specifies the name of a procedure. The job to be renewed has to deal with this procedure.

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

task

Specifies the type of task to be performed by the item procedure.

The procedure to be recovered has to deal with a procedure of this task type.

install

An installation procedure.

configure

A configuration procedure.

activate

An activation procedure.

uninstall

An uninstallation procedure.

The parameter is ignored, if cname is coded.

timeout

Specifies the timeout specifies the expiration period of a job in the form “h” where h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
timeout should not be coded with the jobTimeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 168

version

Specifies the version of the item being deregistered.

userJobMessage

Specifies the custom administrator message for a renewed job in a job container.

userMessage

Specifies the custom administrator message for a job container.

Note: The first job in Error Status in the job list, that matches the item, version, procedure, and installedWith parameters, will be selected for the renew operation, which uses the procedure specified by the procedure parameter.

Note: In the "cname" -variant of the command the parameter block item, version and procedure are optional. If they are not coded, all jobs of the container that can be renewed will be renewed.

resume—Resume a halted Distribution

This command is used to resume a halted distribution.

It has the following format:

```
resume name=jobcontainer_name
    [sendtime="yyyy-mm-dd hh:mm" ]
    [halttime="yyyy-mm-dd hh:mm" ]
```

halttime

Specifies the date and time at which a distribution order should be halted. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm".

Note: Max_Dist_Time determines the maximum number of minutes that can elapse before a container distribution is set to an error state.

name

Specifies the name of the distribution that is halted and has to be resumed.

sendtime

Specifies the time to send the associated distribution to the addressed domain managers. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm". If the parameter is not coded, the actual date and time is used.

rregproc—Registering Procedures Remotely

This command is used to send an order to register a procedure in the library of one or more domain managers. The command is given to the enterprise manager and directed to a domain or a domain group.

The command is valid for enterprise manager only.

It has the following format:

```
rregproc  item=item_name
    version=version_name
    procedure=procedure_name
    {{area=area_name} | {domain=domain_name} | toAllAreas}
    [cname=name]
    [sendTime="YYYY-MM-DD hh:mm"]
    [haltTime="YYYY-MM-DD hh:mm"]
```

area

Specifies the name of a CA ITCM domainwhere the procedure is to be registered at the specified product.

You can specify this parameter more than once to address a list of CA ITCM domains.

This parameter should not be coded with the domain or toAllAreas parameters.

cname

Specifies an optional unique name of a distribution container.

If no cname parameter is given, a name is generated internally and the command line interface returns this identifier.

domain

Specifies the name of a CA ITCM domain group where the procedure is to be registered at the specified product.

You can specify this parameter more than once to address a list of CA ITCM domain groups.

This parameter should not be coded with the area or toAllAreas parameters.

haltTime

Specifies the date and time at which a distribution order should be halted. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm".

item

Specifies the name of the item where the specified procedure will be registered.

procedure

Specifies the name of a procedure to be registered.

sendTime

Specifies the time to send the associated distribution to the addressed domain managers. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm". If the parameter is not coded, the actual date and time is used.

toAllAreas

If coded the related distribution will be sent to all CA ITCM domains registered at the enterprise manager.

"toAllAreas" should not be coded with the "area" or "domain" parameters.

version

Specifies the version of the item at which the specified procedures will be registered.

Example:

Consider the following syntax example, as entered from a Windows server:

```
cadsmcmd rregproc item="Software TestC" version=0.46 procedure=Install2
area=N_LOCAL_SRV_02 area=WO_LOCAL_SRV cname=TestCont2 sendtime="2005-03-04 17:35"
halttime="2005-03-04 22:55"
```

rregsw—Registering Software Remotely

The rregsw command is used to distribute a new software program from the enterprise manager library to a named list of domains or domain groups, and to register it in the library of the domain managers at these domains.

The rregsw command is valid only when issued from the enterprise manager command line.

This command has the following format:

```
rregsw  item=item_name
    version=version_name
    {{area=area_name} | {domain=domain_name} | toAllAreas}
    [cname=name]
    [sendTime="YYYY-MM-DD hh:mm"]
    [haltTime="YYYY-MM-DD hh:mm"]
    [delta[={y|n}]]
```

area

Specifies the name of a CA ITCM domainwhere an item is to be registered.

You can specify this parameter more than once to address a list of CA ITCM domains.

This parameter should not be coded with the domain or toAllAreas parameters.

cname

Specifies the optional, unique name of the distribution.

If no cname parameter is given, a name is generated internally and the command line interface returns this identifier.

delta

Indicates that if the item is based on a previous version, then only the delta should be sent to remote CA ITCM domains. The previous version must already be registered at the remote CA ITCM domains.

domain

Specifies the name of a CA ITCM domain group where the specified item will be registered.

You can specify this parameter more than once to address a list of CA ITCM domain groups.

This parameter should not be coded with the area or toAllAreas parameters.

halttime

Specifies the date and time at which a distribution order should be halted. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm".

For more information about resuming a halted distribution, see the resume command.

item

Specifies the name of the item being deregistered.

sendtime

Specifies the time to send the associated distribution to the addressed domain managers. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm". If the parameter is not coded, the actual date and time is used.

toAllAreas

If coded the related distribution will be sent to all CA ITCM domains registered at the enterprise manager.

"toAllAreas" should not be coded with the "area" or "domain" parameters.

version

Specifies the version of the item being registered.

Example:

```
cadsmcmd rregsw item="Software TestC" version=0.46 area=N_LOCAL_SRV
area=N_LOCAL_SRV_02 area=W0_LOCAL_SRV
cname=TestCont1 sendtime="2005-03-03 17:05" halttime="2005-03-03 21:05"
```

scalabilityServer

This section contains the following topics:

collectSector—Collect Sector Information at Scalability Server (see page 238)

collectSector—Collect Sector Information at Scalability Server

Using this action, you schedule a request for collecting the sector information at the specified scalability server.

It has the following syntax:

```
scalabilityServer action=collectSector
    name=name_of_scalability_server
    [COLLECTALL]
```

name

Specifies the name of the scalability server from which the sector information is collected.

COLLECTALL

(Optional) This option drives the amount of data collected from the sector.

If not coded only the information about common computers and users is collected.

If coded the entire available sector information - including asset management inventory data - is collected.

Note: You should be aware of the following:

- The action works asynchronously. It returns as soon as the request is scheduled. It does not wait for any engine to complete this task.
- The name of the scalability server has to be unique; otherwise the action will fail.
- The addressed asset management manager automatically determines the engine responsible for processing this request from the name of the scalability server.
- The action is provided for a domain manager only. When launched for an enterprise manager it is rejected with error.

security

Use the security action to manage the security area assignments for some types of distributed objects by post-processing.

The command is available for domain managers.

This section contains the following topics:

linkSAToSO

The command links a list of specified secured objects to a specified security area.

This command is valid for domain managers only.

It has the following format:

```
cadsmcmd security       action=linkSAToSO
    name=security_area_name
    {soName=secured_object_name}
    classId=security_class_identifier
    [folderName=folder_name
    folderClassId=identifier_of_the_folder's_class]
    [domainName=domain_name]
```

name

Specifies the name of the security area to that the specified objects are linked to.

soName

Specifies the name (label) of the secured object to be linked to the specified security area.

The option can be coded more than once for passing a list of secured objects.

If the secured object is of security class "SoftwarePackage," you have to specify the name and the version in the following format:

`«name»|«version».`

If «name» contains blanks, you have to enclose «name»|«version» in quotation marks: `"«name»|«version»".`

Example:

name testPackage and version 1.0
`soname=testPackage|1.0`

name CA ITCM Manager and version 12.0

soname="CA ITCM Manager|12.0"

classId

Specifes the identifier of the security class to which the secured objects belong. The following identifiers are valid:

SoftwarePackage

for the software package class

SoftwareJob

for the software job class

SoftwareJobContainer

for the software job container class

SoftwarePolicy

for the class of software based policies.

folderName

Specifies the name of a folder or container to which all the specified secured objects belong. The option is mandatory for the following security classes:

SoftwareJob

In any other case the parameter is ignored.

folderClassId

Specifies the identifier of the security class to which the folder specified belongs. The following identifiers are valid:

- SoftwareJobContainer
- SoftwarePolicy

The parameter is ignored whenever the parameter folderName is ignored.

domainName

Specifies the name of the domain to which all secured objects specified and possible folders belong. If the parameter is not coded then it is assumed that all objects belong to the local domain of the manager addressed by this CADSMCMD call.

Note: This option is obsolete for security objects of type SoftwarePackage and is ignored when coded.

cadsmcmd.example.linkSAToSo

The following example shows how to link the secured object "RAC:srktest" to the security area "m1_area001".

```
Cadsmcmd security action=linkSAToSO name=m1_area001 soname="RAC:srktest [1/8/2010
12:55PM]" classid=softwarejobcontainer domainname=testdomain
```

unlinkSAFromSO

The command unlinks a list of specified secured objects from a specified security area.

This command is valid for domain managers only.

It has the following format:

```
cadsmcmd security       action=unlinkSAFromSO
    name=security_area_name
    {soName=secured_object_name}
    classId=security_class_identifier
    [folderName=folder_name
    folderClassId=identifier_of_the_folder's_class]
    [domainName=domain_name]
```

name

Specifies the name of the security area that the specified objects are unlinked from.

soName

Specifies the name (label) of the secured object to be unlinked from the specified security area.

The option can be coded more than once for passing a list of secured objects.

If the secured object is of security class "SoftwarePackage," you have to specify the name and the version in the following format:

`«name»|«version».`

If «name» contains blanks, you have to enclose «name»|«version» in quotation marks: `"«name»|«version»".`

Example:

name testPackage and version 1.0
`soname=testPackage|1.0`

name CA ITCM Manager and version 12.0

soname="CA ITCM Manager|12.0"

classId

Specifes the identifier of the security class to which the secured objects belong. The following identifiers are valid:

SoftwarePackage

for the software package class

SoftwareJob

for the software job class

SoftwareJobContainer

for the software job container class

SoftwarePolicy

for the class of software based policies.

folderName

Specifies the name of a folder or container to which all the specified secured objects belong. The option is mandatory for the following security classes:

SoftwareJob

In any other case the parameter is ignored.

folderClassId

Specifies the identifier of the security class to which the folder specified belongs. The following identifiers are valid:

- SoftwareJobContainer
- SoftwarePolicy

The parameter is ignored whenever the parameter folderName is ignored.

domainName

Specifies the name of the domain to which all secured objects specified and possible folders belong. If the parameter is not coded then it is assumed that all objects belong to the local domain of the manager addressed by this CADSMCMD call.

Note: This option is obsolete for security objects of type SoftwarePackage and is ignored when coded.

cadsmcmd.example.unlinkSAFromSO

The following example shows how to unlink the secured object "RAC:srktest" from the security area "m1_area001"

```
Cadsmcmd security action=unlinkSAFromSO name=m1_area001 soname="RAC:srktest
[1/8/2010 12:55PM]" classid=softwarejobcontainer domainname=testdomain
```

servergroup—Manage Servergroups

A server group (ServerGroup) is a collection of scalability servers.

All the methods of ServerGroup are available on the domain and enterprise manager until stated otherwise.

This section contains the following topics:

General Group Management

This section contains the following topics:

create—Create Server Groups

This action allows you to create server groups.

It has the following format:

```
servergroup action=create
    name=groupname
    [superGroup=nameofsupergroup]
    [comment=comment]
    [{server=scalability_server} | addall]
    [InheritPerms[={y|n}]]
```

addall

All scalability servers will be added to the group (all that exist at creation time).

comment

Specifies a comment on the group to be created.

inheritPerms

This parameter specifies if the group being created inherits security permissions or not.
If “InheritPerms” or “InheritPerms=y” is coded then the specified group will inherit permissions to it members and becomes a security group.
If “InheritPerms=n” is coded then the specified group will not inherit permissions to its members and therefore is no security group.
Default: InheritPerms=y .

name

Specifies the name of the server group to be created.

server

Specifies the name of a scalability server to be added to the group.

This parameter can be coded more than once to add more than one server to the group.

superGroup

Specifies the name of an existing server group where the group in question will be created as a subgroup.

If not coded the group will be created at the system folder "Scalability Server Groups".

Example

A server group is supposed to be created:

- The group should be named LA_Server.
- It should collect the scalability servers LA_Server01 and LA_Server02.
- The group should initially not be a subgroup of any other group.

The following command will create such a group:

```
cadsmcmd servergroup action=create name=LA_Server server=LA_Server01
server=LA_Server02
```

delete—Delete Server Groups

This action allows you to delete server groups.

This action has the following format:

```
servergroup action=delete name=groupname
```

name

Specifies the name of the server group to be deleted.

Example

To delete the group sg the following command has to be entered:

```
cadsmcmd servergroup action=delete name=sg
```

list—List Server Groups

This action allows you to list all server groups.

This action has the following format:

```
servergroup action=list
```

Example

The following command will generate a list of all server groups:

```
cadsmcmd servergroup action=list
```

modfiy—Modify Server Groups

This action allows you to modify server groups.

It has the following format:

```
servergroup action=modify
    name=groupname
    [NewName=newname]
    [InheritPerms[={y|n}]]
    [comment=comment]
```

comment

Specifies the new comment on the group. To erase a comment enter the empty string ("").

inheritPerms

This optional parameter specifies if the group being modified is a security group or not.
If “InheritPerms=n” is coded then the specified group will not inherit permissions to its members and therefore is no security group any longer.
If “InheritPerms” or “InheritPerms=y” is coded then the specified group will inherit permissions to it members and therefore becomes a security group.
If the parameter is not coded then the permission inheritance settings of the group are not changed.

name

Specifies the name of the server group to be modified.

NewName

Specifies the new name of the server group.

Example

Suppose the group "LA_Server" has to be modified.

The name of the group should be changed to "LA District".

The following command will provide this change:

```
cadsmcmd servergroup action=modify name=LA_Server newName="LA District"
```

showAttr—Show Attributes of Server Groups

This action allows you to list the attributes of server groups.

This action has the following format:

```
servergroup action=showattr
    name=groupname
    [groupScope={global | local}]
```

name

Specifies the name of the server group showing its attributes.

groupScope

Specifies the scope of the group specified. The following values are valid for this option:

Global

The group is a global group, created at an enterprise manager. It may be replicated to a domain manager.

Local

Specifies a local group, created at the domain manager the CADSMCMD is connected to.

If "groupScope" is not coded then the CADSMCMD first tries to locate the group as a local group. If this fails then it is tried to locate the group as a global one.

Example

```
cadsmcmd servergroup action=showattr name=sg_1
```

The output will be as follows:

```
-------------------------------------------------------------------------------
Attributes of the server group "sg_1"
-------------------------------------------------------------------------------
Server group name.....................: sg_1
Comment...............................:
Security type.........................: security group
Creation date.........................: 2006-08-28
Creation time.........................: 09:57
Change date...........................: 2006-08-28
Change time...........................: 09:57

SDCMD<A000000>: OK
```

Membership Management

This section contains the following topics:

add—Add Server to Server Groups

This action allows you to add servers to a server group.

This action has the following format:

```
servergroup action=add
    name=groupname
    {{{server=server_name |
    qserver=(server_name sep domain_name)}}
    | addall}
    [sep=separator_sign]
```

name

Specifies the name of the server group to which the servers will be added.

qserver

Specifies the server as a pair of server name and the name of the domain to which it belongs. By default, the names are separated by a dot (".") but an alternate separator may be specified with the sep option. The separator should not be part of the names.

The qserver can be coded with the server in one call.

This server will be added.

sep

Specifies an alternate separator used with the qserver option. The separator should not be part of the names coded with the qserver options.

server

Specifies the name of a scalability server to be added to the group.

addall

All scalability servers, which exist at creation time, are added to the group.

Note : The parameters for addressing a scalability server (target) can be coded more than once to address more than one scalability server.

Example

To add the servers server01 and server02 to the group sg_1 enter the following command:

```
cadsmcmd servergroup action=add name=sg_1 server=server01 server=server02
```

listMem—List Members of Server Groups

This action allows to list the members of a group.

This action has the following format:

```
servergroup action=listMem
    name=groupname
    [groupScope={global | local}]
```

name

Specifies the name of the group.

groupScope

Specifies the scope of the group specified. The following values are valid for this option:

Global

The group is a global group, created at an enterprise manager. It may be replicated to a domain manager.

Local

Specifies a local group, created at the domain manager the CADSMCMD is connected to.

If "groupScope" is not coded then the CADSMCMD first tries to locate the group as a local group. If this fails then it is tried to locate the group as a global one.

Example

To receive the list of members of group "sg_1", enter the following command:

```
cadsmcmd servergroup action=listMem name=sg_1
```

The following output list will be provided:

```
--------------------------------------------------------------------------------
Members of server group "sg_1" (local)
--------------------------------------------------------------------------------
server01 (server01)
server02 (server02)
```

listMembership—List Membership of Server Groups

This action allows you to list those groups a group is contained in as a subgroup.

This action has the following format:

```
servergroup action=listMembership
    name=groupname
    [groupScope={global | local}]
```

name

Specifies the name of the server group for which the membership will be listed.

groupScope

Specifies the scope of the group specified. The following values are valid for this option:

Global

The group is a global group, created at an enterprise manager. It may be replicated to a domain manager.

Local

Specifies a local group, created at the domain manager the CADSMCMD is connected to.

If "groupScope" is not coded then the CADSMCMD first tries to locate the group as a local group. If this fails then it is tried to locate the group as a global one.

Example

To receive a list of servergroups the following command has to be entered:

```
cadsmcmd servergroup action=listMembership name=sg_2
```

The following information is provided:

```
------------------------------------------------------------------------------
"sg_2" is member of the following Groups
------------------------------------------------------------------------------
sg_1 (2006-08-28 09:57)
sg_x (2006-08-28 10:56)
```

remove—Remove Server from Server Groups

This action allows you to remove servers from the server group.

This action has the following format:

```
servergroup action=remove
    name=groupname
    {{{server=server_name |
    qserver=(server_name sep domain_name)}}
    [sep=separator_sign]
```

name

Specifies the name of the server group from which one or more members are to be removed.

qserver

Specifies the server as a pair of server name and the name of the domain to which it belongs. By default, the names are separated by a dot (".") but an alternate separator may be specified with the sep option. The separator should not be part of the names.

This server will be removed.

sep

Specifies an alternate separator used with the qserver option. The separator should not be part of the names coded with the qserver options.

server

Specifies the name of the server to be removed from the group.

This parameter can be coded more than once to remove more than one server from the group.

Note: If a server of ambiguous name is to be removed from a group then use the qserver option to remove it. Otherwise the result might be unpredictable.

Example

To remove the servers server01 and server02 from the group sg_1 enter the following command:

```
cadsmcmd servergroup action=remove name=sg_1 server=server01 server=server02
```

Hierarchy Management

This section contains the following topics:

addSubgroups—Add Subgroups to Server Groups

This action allows you to add subgroups to a server group.

This action has the following format:

```
servergroup action=addSubgroups
    [name=groupname]
    {servgrp=groupname}
```

name

Name of the server group the server group specified by servgrp will be added to.

If this parameter is not coded the subgroups are copied to the container "Scalability Server Groups".

servgrp

Specifies the name of an existing server group to be added.

This parameter can be coded more than once to add a list of servers.

Example

Suppose the server groups named "LA admin" and "LA sales" should be added to the server group "LA District" as subgroups.

The following command will assign these groups as subgroups to the supergroup "LA District":

```
cadsmcmd servergroup action=addSubgroups name="LA District" servgrp="LA admin"
servgrp="LA sales"
```

By the way, the groups "LA admin" and "LA sales" have to be defined at the time of command invocation, otherwise, an error is reported for the missing subgroup.

listSubgroups—List Subgroups of Server Groups

This action allows you to list the subgroups of a server group.

This action has the following format:

```
servergroup action=listSubgroups
    [name=groupname]
    [groupScope={global | local}]
```

name

Specifies the name of the server group of which the subgroups will be listed.

If this parameter is not coded the server groups of the container "Scalability Server Groups" are listed.

groupScope

Specifies the scope of the group specified. The following values are valid for this option:

Global

The group is a global group, created at an enterprise manager. It may be replicated to a domain manager.

Local

Specifies a local group, created at the domain manager the CADSMCMD is connected to.

If "groupScope" is not coded then the CADSMCMD first tries to locate the group as a local group. If this fails then it is tried to locate the group as a global one.

The output list has the following format:

```
--------------------------------------------------------------------------
List of Subgroups of Server Group "groupname"
--------------------------------------------------------------------------
subgroupname
subgroupname
...
```

Example

To list all subgroups associated with the server group "LA District" code the following:

```
cadsmcmd servergroup action=listSubgroups name="LA District"
```

removeSubgroups—Remove Subgroups from Server Groups

This action allows you to remove subgroups from a server group.

This action has the following format:

```
servergroup action=removeSubgroups
    [name=groupname]
    {servgrp=groupname}
```

name

Specifies the name of the server group the server group specified by servgrp will be removed from.

If this parameter is not coded then the subgroups are removed from the container "Scalability Server Groups".

servgrp

Specifies the name of an existing server group to be removed.

This parameter can be coded more than once to remove a list of server groups.

Note: "Remove" does not mean that any server group of the servgrp parameter is deleted. It is only unlinked from the supergroup specified with the name parameter.

Note: If a server group A contains the server group B, and if B is not contained by any other server group, nor by the container "Scalability Server Groups", then the attempt to remove B from A will result in an error. At least one of these containments must remain.

Example

Suppose that the server groups named "LA admin" and "LA sales" should be removed from the server group "LA District" again.

The following command will remove these groups as subgroups from the supergroup "LA District":

```
cadsmcmd servergroup action=removeSubgroups name="LA District" servgrp="LA admin"
servgrp="LA sales"
```

The groups "LA admin" and "LA sales" are not deleted from the database by the command; they are only removed from the group "LA District". Any other containment relations beside the ones previously shown are still present.

The command will report an error if the containment relation to be broken is the only one for one of the subgroups to be removed. For example, "LA admin" is only mentioned as a subgroup of "LA District" while the other subgroups are also mentioned in the folder "Scalability Server Groups" or in other groups.

stagingserver—scalability server Commands

The scalability server - in previous releases called "staging server" - is used in SD networks to provide software to its attached target computers for installation and to optimize the communication links between the domain managers and its associated target computers. A scalability server is always attached to a unique domain manager.

The command stagingserver is valid for domain managers only.

This section contains the following topics:

activateStageCheck—Activate the Stage Check Program

This actions starts the Stage Check program that initiates a check for pending orders.

It will not wait for its completion.

This action has the following format:

```
stagingserver action=activatestagecheck
    computer=name
```

computer

Specifies the name of the computer where the Stage Check program will be activated.

addItem—Add Item to scalability server

This action adds an item to one or more staging libraries.

This action has the following format:

```
stagingserver action=additem
    item=item_name
    version=item_version
    {{computer=computer_name} | {compgrp=computer_group_name}}
    [cname=name]
    [atTime="YYYY-MM-DD hh:mm"]
    [ timeout=hours ]
    [{nolinkage | transaction | synchronized } ]
    [jcPriority={1,...,10}]
```

attime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

cname

Specifies the the optional unique name of a job container where the generated job will be stored.

If a job container / distribution with the specified name already exists an error will be reported.
If "cname" is not coded, a generic name is generated.

compgrp

Specifies the name of a server group of scalability servers to which the item is to be delivered.

The parameter can be coded more than once to address a list of server groups.

The parameter should not be coded with the computer parameter.

computer

Specifies the name of a scalability server to which the item is to be delivered.

The parameter can be coded more than once to address a list of scalability servers.

The parameter should not be coded with the compgrp parameter.

item

Specifies the name of the item to be delivered to the scalability servers.

jcPriority

Specifies the priority of the job container.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded priority 5 is assumed.

nolinkage

Run the job independently of the other jobs in the container.

synchronized

The job linkage option is set to "Synchronized job execution".

timeout

Specifies the timeout specifies the expiration period of a job in the form “h” where h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
timeout should not be coded with the jobTimeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 168

transaction

The job linkage option is set to "Batch job execution".

version

Specifies the version of the item to be added.

list—List scalability servers

A list of all scalability servers is shown. An additional filter can be specified to restrict the list to only those scalability servers that fulfill the filter criteria. If no filters are given the complete list of products is shown. The filter is limited to100 bytes.

This action has the following format:

```
stagingserver action=list
    [{filter="filter"
    | filterfile=file_name}]
```

filter

Specifies the expression to confine the amount of information listed.

You can also use the attributes that are shown by the "targetcomputer list" action.

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command.

Example

To generate a list staging servers of all server groups, enter the following command:

```
cadsmcmd stagingserver action=list
```

The command provides an output list on stdout of the following format:

```
--------------------------------------------------------------------------------
List of staging servers
--------------------------------------------------------------------------------
Test01mc
Test02mc
WWWWH01
--------------------------------------------------------------------------------
Total number of computers: 3
Number of staging servers found: 3
Number of staging servers shown: 3
SDCMD<A000000>: OK
```

To generate a list staging servers of all server groups starting with "Test" using filters, enter the following command:

```
cadsmcmd stagingserver action=list filter="computer name=Test*"
```

The output on stdout is as follows:

```
--------------------------------------------------------------------------------
List of staging servers
--------------------------------------------------------------------------------
Test01mc
Test02mc
--------------------------------------------------------------------------------
Total number of computers: 3
Number of staging servers found: 3
Number of staging servers shown: 2
SDCMD<A000000>: OK
```

listComp—List Computers Assigned to scalability server

This actions displays a list of all computers that are assigned to the specified scalability server. The list only contains the name of the computers.

This action has the following format:

```
stagingserver action=listcomp
    name=staging_server_name
```

name

Specifies the name of the scalability server for which the assigned computers will be listed.

listItem—List Items of scalability server

This action show items available at specified scalability server. The list shows the names of the items and their versions.

This action has the following format:

```
stagingserver action=listitem
    name=staging_server_name
```

name

Specifies the name of the scalability server for which the associated items will be shown.

removeItem—Remove Item from Scalability Server Libraries

This action removes an item from one or more staging libraries.

This action has the following format:

```
stagingserver action=removeitem
    item=item_name
    version=item_version
    { { computer=computer_name }
    | { compgrp=computer_group_name } }
    [cname=name]
    [{nolinkage | transaction | synchronized } ]
    [atTime="YYYY-MM-DD hh:mm"]
    [ timeout=hours]
    [jcPriority={1,...,10}]
```

attime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

cname

Specifies the the optional unique name of a job container where the generated job will be stored.

If a job container / distribution with the specified name already exists an error will be reported.
If "cname" is not coded, a generic name is generated.

compgrp

Specifies the name of a server group of scalability servers from which the item is to be removed.

The parameter can be coded more than once to address a list of server groups.

The parameter should not be coded with the computer parameter.

computer

Specifies the name of the scalability server from which the item is to be removed.

The parameter can be coded more than once to address a list of scalability servers.

This parameter should not be coded with the compgrp parameter.

item

Specifies the name of the item to be removed.

jcPriority

Specifies the priority of the job container.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded priority 5 is assumed.

nolinkage

Run the job independently of the other jobs in the container.

synchronized

The job linkage option is set to "Synchronized job execution".

timeout

Specifies the timeout specifies the expiration period of a job in the form “h” where h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
timeout should not be coded with the jobTimeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 168

transaction

The job linkage option is set to "Batch job execution".

version

Specifies the version of the item to be removed.

Note: If neither "synchronized" nor "transaction" is coded, then a job container of type "nolinkage" will be created.

swlibrary—Software Library Commands

All methods of this chapter have been provided on enterprise and domain manager, unless something else is stated.

The software package library (swlibrary) administers all the software and groups of software and procedures an enterprise and domain manager deals with.

This section contains the following topics:

General Software Group Management

This section contains the following topics:

assignGroupToTarget—Assign Group to Target

This action lets you assign a software group or a procedure group to target computers or computer groups (including software templates). If the software group contains subgroups, the subgroups take part in the assignment recursively.

Besides other information, the name of the job container or distribution used is printed on standard output.

This action has the following format:

```
swLibrary action=assignGroupToTarget name=group_name
    {{computer=computer_name} |
    {compgrp=computer_group_name | ccompgrp=(computer_group_name sep groupScope)}
    [sep=separator_sign]
    | {swTemplates=software_template_name}}
    cname=[name]
    [dname=distribution_name]
    [unsealed]
    [{nolinkage|transaction[rollback]|synchronized}[nocascade]]
    [deliveryTime="YYYY-MM-DD hh:mm"]
    [atTime="YYYY-MM-DD hh:mm"]
    after={exacttime|boottime}
    [calendarname=delivery_calendar]
```

[preaction={none|reboot|logoff}]
[postaction={none|reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}]
[promptUser[={y|n}]]
[allowCancel[={y|n}]]
[execTimedOut[={y|n}]]
[prompt=*days.hours*]
[offline[={y|n}]]
[runAtShutdown[={y|n}]]
[preventLogon[={y|n}]]
[globalTime]
[nocalendar]
[resolveQuery]
[parameters=*user_parameters*]
[repeat={y|n}]
[stagingserver]
[jobTimeout=*d.h*]
[procedures=*procedures_file*]
[{{area=area_name} | {domain=domain_name} | toAllAreas}]
[sendTime="*YYYY-MM-DD hh:mm*"]
[haltTime="*YYYY-MM-DD hh:mm*"]
[jcPriority={1,...,10}]
[distPriority={1,...,10}]
[reinstall[={y|n}]]

after

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

allowCancel

If "allowCancel" or "allowCancel=y" is coded, the user will be granted to cancel the job's execution.
If "allowCancel=n" is coded, the user will not be granted.

If the parameter is not coded, the default is given by the related procedure's job option value.

If neither promptUser is set by this call nor already enabled, then this parameter is ignored. No warning is given.

area

Specifies the name of a CA ITCM domain the distribution will be sent to.

The parameter can be coded more than once to specify a list of CA ITCM domains.

If this parameter is coded then the distribution will be sealed and automatically sent to its destinations after the command has been successfully completed.

If neither the area nor the domain nor the toAllAreas parameter is coded, the distribution remains unsealed and can further be manipulated.

This parameter should not be coded with the domain or toAllAreas parameters.

(enterprise manager only)

atTime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

calendarname

Specifies the name of the calendar that controls the time when the evaluation can take place.

ccompgrp

Specifies the computer group to be addressed as a pair of group name and scope. By default, name and scope are separated by a dot ("."), but an alternate separator can be specified with the "sep" option. The separator should **not** be used as part of the names.

For the group scope only the following values are valid.

local

The group is created at the local domain manager,

global

The group has been created at the enterprise manager and may be replicated.

The "ccompgrp" can be coded with the "compgrp" in one call.

cname

Specifies the the optional unique name of a job container.

If no cname parameter is given, a name is generated internally and the command line interface returns this identifier.

The generated jobs will be assigned to it.

If this job container does not exist, it will be created.

If it exists, then the generated jobs will be added - if the container is unsealed.

If it is sealed, cadsmcmd terminates with an error.

compgrp

Specifies the computer group to be addressed.

The parameter can be coded more than once to address more than one computer group.

This parameter should not be coded with the computer or swTemplates parameters.

computer

Specifies the name of a computer to be addressed.

The parameter can be coded more than once to address more than one computer.

This parameter should not be coded with the compgrp or swTemplates parameters.

(domain manager only)

deliveryTime

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

distPriority

Specifies the priority that will be assigned to the distribution.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded priority 5 is assumed.

dname

Specifies the name of the distribution the jobs and job container will be stored in.

If the distribution already exists, then this distribution is used - if it is still open for manipulations, otherwise an error message is launched. If the distribution does not exist, it will be created.

If the parameter is not coded, a distribution of default name is generated.

(enterprise manager only)

domain

Specifies the name of a CA ITCM domain group the distribution will be sent to.

The parameter can be coded more than once to address a list of CA ITCM domain groups.

If this parameter is coded then the distribution will be sealed and automatically sent to its destinations after the command has been successfully completed.

If neither the area nor the domain nor the toAllAreas parameter is coded, the distribution remains unsealed and can further be manipulated.

This parameter should not be coded with the area or toAllAreas parameters.

(enterprise manager only)

execTimedOut

If "execTimedOut " or "execTimedOut=y" is coded, the job's execution will be automatically started when the user prompt times out.
If "execTimedOut=n", the job will not automatically be started.

If the parameter is not coded, the default is given by the related procedure's job option value.

If neither promptUser is set by this call nor is already enabled, then this parameter is ignored. No warning is given.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time. (default)

haltTime

Specifies the date and time at which a distribution order should be halted. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm".

For more information about resuming a halted distribution, see the resume command.

jcPriority

Specifies the priority of the job container.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded the priority is not changed.

jobTimeout

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

name

Specifies the name of the group. If it is a software group the members of the group will be installed at the targets site.

The procedure of the members to run at the target can be specified and configured with the procedures option. If it is not coded the default install procedures of the members are taken.

If it is a procedure group the contained procedures will be executed at the target site.

noCalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

If the parameter is not coded, the default is given by the related procedure's job option value.

noCascade

The job linkage option is set to "Ignore cascading".

This parameter is allowed only if "transaction" or "synchronized" are coded.

noLinkage

Run the job independently of the other jobs in the container.

offline

If "offline" or "offline=y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.
If "offline=n" the connection will not be released.

If the parameter is not coded, the default is given by the related procedure's job option value.

parameters

Specifies the user parameters for the procedure.
Multiple user parameters must be separated by spaces inside double quotes.
If an empty string is coded, then no parameters are assumed.

If the parameter is not coded, the default is given by the related procedure's job option value.

postaction

Specifies the necessary actions that must take place after completion of the specified procedure. The following values are valid:

none

Performs no post-action.

reboot

Restarts the system after completion of the procedure.

logoff

Logs off the user after completion of the procedure.

rebootAtEnd

Restarts the system after completion of all the jobs of this container.

logoffAtEnd

Logs off the user after completion of all the jobs of this container.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

If the parameter is not coded, the default is given by the related procedure's job option value.

preaction

Specifies that necessary actions must take place before the specified procedure starts. The following values are valid:

none

Performs no pre-action.

reboot

Restarts the system before the start of the procedure.

logoff

Logs off the user before the start of the procedure.

If the parameter is not coded, the default is given by the related procedure's job option value.

preventLogon

If "preventLogon" or "preventLogon=y" is coded, then the user logon will be rejected during the job's runtime. If a user is already logged on, the execution of the job is delayed until the user logs off.
If "preventLogon=n" the user logon will not be rejected.

If the parameter is not coded, the default is given by the related procedure's job option value.

procedures

Specifies the name of the file defining the procedures to be used for the software group and / or to define individual job options for the procedures (software and procedure groups).

The content of this procedures file is shown in Procedures File Format (see page 272).

prompt=d.h

Specifies the time period in which the user will be prompted for the job's start.

The period's format is "*d.h*" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0). If the specified value falls below the valid minimum, the value is replaced by the minimum. If the specified value exceeds the valid maximum, then the value is replaced by the maximum. No warning is given.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

If the parameter is not coded, the default is given by the related procedure's job option value.

reinstall

The "reinstall" option is optional, the default is "reinstall=n".

If "reinstall" or "reinstall=y" is coded then a possible installation record related to the order is removed before the job is evaluated by the domain manager.

If "reinstall=n" is coded then such records will remain.

Note: Option only applies to install jobs. In any other cases it is ignored.

resolveQuery

Evaluate a possibly related query group before the job is generated.

rollback

Set the job linkage option to "Enable Transaction".

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

sendtime

Specifies the time to send the associated distribution to the addressed domain managers. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm". If the parameter is not coded, the actual date and time is used.

(Enterprise manager only)

sep

Specifies an alternate separator used with the "ccompgroup" option. The separator should not be part of the group name code with the "ccompgroup" options.

stagingServer

Deliver the related product also to the staging libraries of the scalability servers of the target computers.

This parameter applies to procedures of task type "install" only!

swTemplates

Specifies the name of the software template.

The parameter can be coded more than once to address more than one software template.
If it is coded, then the cname parameter becomes obsolete, the jobs to be generated are not assigned to any job container but to the software template. To do so, the software templates of the list should not be sealed.

This parameter should not be coded with the computer or compgrp parameters.

synchronized

The job linkage option is set to "Synchronized job execution".

toAllAreas

If coded the related distribution will be sent to all CA ITCM domains registered at the enterprise manager.

"toAllAreas" should not be coded with the "area" or "domain" parameters.

If neither the area nor the domain nor the toAllAreas parameter is coded, the distribution remains unsealed and can further be manipulated.

(enterprise manager only)

transaction

The job linkage option is set to "Batch job execution".

This is the default.

unsealed

The addressed job container remains unsealed after the generated job has successfully been created.

(domain manager only)

Notes:

If neither area, nor domain, nor toAllAreas is coded, the distribution container remains unsealed and further manipulation is possible.

The parameters "allowCancel," "execTimedOut" and "prompt" are valid, if the "promptUser" option is already set or is set by this command using the "promptUser" parameter.
If neither the "promptUser" option of a job nor the "promptUser" parameter of the command is set, these parameters will be ignored.
No warning is given.

Procedures File Format

The content of the procedures file consists of entries of the following syntax:

```
item = item
version = version
procedure = procedure
task={install|configure|activate|uninstall}
[deliverytime="YYYY-MM-DD hh:mm" ]
[attime="YYYY-MM-DD hh:mm" ]
[after={exacttime|boottime}]
[calendarname=deliverycalendar]
[preaction={none|reboot|logoff}]
[postaction={none|reboot|logoff|rebootAtEnd|logoffAtEnd}]
[promptUser={y|n}]
[allowCancel={y|n}]
[execTimedOut={y|n}]
[prompt=days.hours]
[offline={y|n}]
[RunAtShutdown={y|n}]
[PreventLogon={y|n}]
[GlobalTime={y|n}]
[nocalendar={y|n}]
[ResolveQuery={y|n}]
[Parameters=userparameter]
[stagingserver={y|n}]
[JobTimeout=d.h]
```

item

Name of the group item this entry is related to

version

Version for the group item this entry is related to

procedure

Specifies the item procedure to be performed

task

Specifies the procedure type.

The following types are valid:

install

The procedure installs the item.

configure

The procedure configures the item

activate

The procedure activates the item.

uninstall

The procedure uninstalls the item.

If the parameter is not coded, install is assumed.

The parameters "item", "version", "procedure", and "task" have to be coded at each line as the first parameters and they have to be coded in the order as in the syntax definition of cadsmcmd swlibrary action=assigngrouptotarget shown. The remaining parameter can be coded in any order.

The meaning of the other parameters corresponds to the meaning of the command parameters shown in cadsmcmd swlibrary action=assigngrouptotarget.

The procedure options for a job procedure are taken from the settings in this list.

If for a procedure an option is not coded in the related list entry then it is taken from the settings on command layer.

If it is not coded there then it is taken from the procedure options registered.

Example

The attributes specified with the command serve as defaults for all jobs to be generated. While for a procedure group it is clear which procedure to run for software groups, this is open. As default for the software group products, the command selects the default install routine of the products for the jobs. All these defaults can be overwritten by additional specifications in the "procedures" file.

Assume a "procedures" file is specified. For each entry of a procedure group it will be checked whether the procedure and product also occurs in the specified "procedures" file or not. If not the commands defaults will be used to generate the job, otherwise the specifications of the first matching "procedures" file entry overwrite the defaults and these modified attributes are used to generate the job. The jobs will be generated in the order the procedures are sorted in the group.

For example, assume a domain manager, the following procedure group and a command as follows:

```
pg:
........Procedure:........configure_001
................Product........test_001
................Version........1.0
........Procedure:........activate_002
................Product........test_002
................Version........1.1

cadsmcmd swlibrary action=assignGroupToTarget name=pg computer=tar01
after=exacttime prompUser allowCancel Parameters="-xon xf -rs"
procedures=c:\temp\procFile.txt
```

Assume further that c:\temp\procFile.txt has the following contents:

```
Item=test_002 version=1.1 procedure=activate_002 task=activate after=boottime
allowCancel=n GlobalTime=y Parameters=""
```

Two jobs will be generated. The first one is for the configure_001 and the attributes of the command are used as defaults, in other words, after=exacttime, promptUser, cancelAllow, and Parameters="-xon xf -rs" . The second job generated is for activate_002, and in this case the defaults from the command are used too, but some of them are overwritten by the related "procedures" file entry. This results in a job with the attributes after=boottime, promptUser, and GlobalTime while allowCancel is not set and Parameters is empty.

In case of a software group, it is only checked if the product matches an entry in the "procedures" file. The first entry found is used. The specified procedure of the matching entry and the specified parameters overwrite the defaults and these modified attributes are used to generate the job.

For example, assume a domain manager, the following software group and a command as follows:

```
swg:
........Product:........test_011
................Version........2.1
................Default installation procedure........inst_011
........Product:........test_012
................Version........2.2
................Default installation procedure........inst_012

cadsmcmd swlibrary action=assignGroupToTarget name=swg computer=tar01
after=exacttime promptUser allowCancel Parameters="-xon xf -rs" preaction=logoff
procedures=c:\temp\procFile.txt
```

Assume further that c:\temp\procFile.txt has the following contents:

```
Item=test_012 version=2.2 procedure=activate_012 task=activate after=boottime
allowCancel=n GlobalTime=y Parameters="-cleanUp"

Item=test_011 version=2.0 procedure=activate_011 task=activate allowCancel=n
GlobalTime=y Parameters="-format"
```

Two jobs will be generated. The first one is for the product test_011 of Version 2.1 and its an installation job using the default procedure inst_011. The job is generated with the command defaults only, because there is no matching entry in the "procedures" file, although the product test_011 is refered to in the file but of a different version. Therefore the job is generated with the attributes after=exacttime, promptUser, allowCancel, and Parameters="-xon xf -rs" .

The second job is generated for the product test_012 of Version 2.2. This time there is a matching entry in the "procedures" file, therefore the defaults given by the command where updated by those of the entry, and the job created is for the activation procedure activate_012 with attributes after=boottime, promptUser, GlobalTime, preaction=logoff, and Parameters="-cleanUp," while allowCancel is not set.

A software group may have a hierarchical structure, in other words, it contains other software or procedure groups. The command works recursively, in other words, it also evaluates the subgroups and generates jobs for their entries. Using hierarchical group structures the user should take into account that the products and procedures are unique in this hierarchy. For example, consider the following structure:

```
swg: (software group)
........Product:........test_001
................Version........2.1
................Default installation procedure........inst_001
........Product:........test_022
................Version........1.1/00
................Default installation procedure........inst_022
swg_1: (software group)
........Product........test_010
................Version........1.0
................Default installation procedure........inst_010
........Product:........test_001
................Version........2.1
................Default installation procedure........inst_001
pg_11: (procedure group)
........Procedure:........configure_001
................Product........test_001
................Version........2.1
........Procedure:........activate_022
................Product........test_022
................Version........1.1/00
pg_1: (procedure group)
........Procedure:........configure_022
................Product........test_022
................Version........1.1/00
........Procedure:........activate_022
................Product........test_022
................Version........1.1/00
```

Invoking the command for this group will cause some warnings and some products and procedures will be ignored. This holds for the product test_001 of Version 2.1 in swg_1 because it already appears at level swg as well as for the procedure activate_022 of product test_022 of Version 1.1/00 in pg_11, because it already appears at pg_1. If a "procedures" file is coded with the command the procedure groups are checked for matching entries first. After they are all completed the software groups of the hierarchy are checked against the remaining entries of the "procedure" files, in other words, against those entries not already used within procedure groups.

For example, consider the swg structure previously shown and the following command on a domain manager:

```
cadsmcmd swlibrary action=assignGroupToTarget name=swg computer=tar01
after=exacttime promptUser allowCancel Parameters="-xon xf -rs"
procedures=c:\temp\procFile.txt
```

Assume further that c:\temp\procFile.txt has the following contents:

```
Item=test_001 version=2.1 procedure=configure_001 task=configure allowCancel=n
Parameters="-init" preaction=reboot

Item=test_001 version=2.1 procedure=install_min task=install allowCancel=n
Parameters="-x -y -z" preaction=logoff postaction=reboot

Item=test_022 version=1.1/00 procedure=configure_022 task=configure
preaction=logoff Repeat noCalendar

Item=test_022 version=1.1/00 procedure=activate_022 task=activate preaction=logoff
Repeat noCalendar
```

This will generate the subsequent jobs with the listed attributes:

```
test_001 2.1: install_min
task=install
after=exacttime
promptUser
Parameters="-x -y -z"
preaction=log0ff
```

```
postaction=reboot
test_022 1.0/00:inst_002
task=install
after=exacttime
promptUser
allowCancel
Parameters="-xon xf -rs"
test_022 1.0/00:configure_022
task=configure
after=exacttime
preaction=logoff
promptUser
allowCancel
Repeat
noCalendar
Parameters="-xon xf -rs"
test_022 1.0/00:activate_022
task=activate
after=exacttime
preaction=logoff
promptUser
allowCancel
Repeat
noCalendar
Parameters="-xon xf -rs"
test_010 1.0:inst_010
task=install
after=exacttime
promptUser
allowCancel
Parameters="-xon xf -rs"
test_001 1.0/00:configure_001
task=configure
after=exacttime
```

```
promptUser
Parameters="-init"
preaction=reboot
```

It will test_001 of version 2.1 in swg_1 as well as activate_022 of test_022 and version 1.1/00 in pg_11 will be ignored.

The entries of the procedures file have the following format:

```
item=item

version=version

procedure=procedure task={install|configure|activate|uninstall}
[deliverytime="YYYY-MM-DD hh:mm" ]
[attime="YYYY-MM-DD hh:mm" ]
[after={exacttime|boottime}] [calendarname=deliverycalendar]
[preaction={none|reboot|logoff}]
[postaction={none|reboot|logoff|rebootAtEnd|logoffAtEnd}]
[promptUser={y|n}]
[allowCancel={y|n}]
[execTimedOut={y|n}]
[prompt=days.hours]
[offline={y|n}]
[RunAtShutdown={y|n}]
[PreventLogon={y|n}]
[GlobalTime={y|n}]
[nocalendar={y|n}]
[ResolveQuery={y|n}]
```

```
[Parameters=userparameter]
[repeat={y|n}]
[stagingserver={y|n}]
[JobTimeout=d.h]
```

item

Name of the group item this entry is related to.

version

Version for the group item this entry is related to.

procedure

Specifies the item procedure to be performed.

task

Specifies the procedure type. The following types are valid:

install

The procedure installs the item.

configure

The procedure configures the item.

activate

The procedure activates the item.

uninstall

The procedure uninstalls the item.

If the parameter is not coded, install is assumed.

The item, version, procedure, and task parameters have to be coded at each line as the first parameters, and they have to be coded in the order as in the syntax definition previously shown. The remaining parameters can be coded in any order.

The other parameters' meanings correspond to the command parameters previously shown. If a y is coded, then the parameter will be set; otherwise, it will not be set. For parameters not set, the default is taken from each command's parameter setting.

Consider a software group swg_1 that should be distributed to the targets given by the group q_grp. The swg_1 shows the following contents and structure:

```
cadsmcmd swlibrary action=listswg name=swg_1

CA IT Client Manager r12

ITCM Command Line Version 12.8.0.xxxx

Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "default manager" as user "<default user>" ...OK.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM AM

--------------------------------------------------------------------------------
List members of procedure group "swg_1"
--------------------------------------------------------------------------------
InstallActivateConfigureUninstall 1 <Generic>, <Regular>, 3(0), schbe05,
Pass the exitcode that should be returned as a user parameter.
TransferComplexDirectoryStructure 1 <Generic>, <Regular>, 329(0), QA Mch schir03,
SD Package with complex directory structure.

SDCMD<A000000>: OK
```

Suppose the requirements for the job container to be used are as follows:

- The jobs are executed in sequence. If a job fails no further job is executed, already executed jobs are rolled back.
- Dependencies should be resolved and install cascades are to be built.

The defaults for all jobs to be generated are:

- The delivery time is 2005-06-30 1:00 pm.
- The processing start time at the targets is 2005-07-05 01:00 am. The processing can start immediately after the timestamp is reached.
- A possible query for the computer group has to be evaluated before the jobs are generated.
- For the remaining attributes the defaults are used.
- Individual attributes for the jobs are passed with the procedure file "c:\temp\proc.usd".

The individual requirements for jobs and items are as follows:

- For the procedures of the procedure group pg_1
- remdir and setdir of tstbase require a reboot after installation.
- inst of tstkpk01 does not have any further requirements.
- For the items of the software group swg_1
- The tstbase is installed by the install procedure "inst".
- A logged in user is logged off before the installation starts.
- Any login attempt during installation is rejected.
- DMS Editor to be installed by the default procedure "Local Install".
- No further, individual requirements.

To distribute and run the associated jobs, the command assignGroupToTarget is used. Due to the individual requirements for some jobs, a procedure file is needed to describe those individual requirements. This procedure file has the following contents:

```
item=tstbase version=1.0 procedure=remdir task=configure postaction=reboot

item=tstbase version=1.0 procedure=setdir task=configure postaction=reboot

item=tstbase version=1.0 procedure=inst task=install preaction=logoff preventLogon
```

The following command will create a job container of default name, populate it with jobs required and schedule it for execution:

```
cadsmcmd swlibrary action=assignGroupToTarget name=swg_1 compgrp=q_grp transaction
rollback deliveryTime="2005-06-30 13:00" atTime="2005-07-05 01:00" after=exacttime
resolveQuery procedures=c:\temp\proc.usd
```

```
CA IT Client Manager r12

ITCM Command Line Version 12.8.0.xxxx

Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "wwwww01b" as user "<default user>" ...OK.

A container with the name q_grp [04/11/02 10:44:10.093] was created.
Evaluate group swg_1
Evaluate group swg_1/pg_1
Evaluate procedures file c:\temp\proc.usd
Generate job "tstbase 1.0:inst" : OK.
Generate job "DMS Editor 6.01.0000:Local Install" : OK.
Generate job "tstbase 1.0:remdir" : OK.
Generate job "tstbase 1.0:setdir" : OK.
Generate job "tstkpk01 1.0:inst" : OK.

SDCMD<A000000>: OK
```

The following command shows the job container generated:

```
cadsmcmd jobContainer action=showAttr name="q_grp [04/11/02 10:44:10.093]"

CA IT Client Manager r12

ITCM Command Line Version 12.8.0.xxxx

Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "wwwww01b" as user "<default user>" ...OK.

------------------------------------------------------------------------------
Show attributes of the job container "q_grp [04/11/02 10:44:10.093]"
------------------------------------------------------------------------------
Job container name.....................: q_grp [04/11/02 10:44:10.093]
Job container properties...............: 28
.....................................Transaction
.....................................Rollback previous jobs on failure
.....................................Cascade install
Sealed.................................: Yes
Created at.............................: 2005-04-11 10:45
Changed at.............................: 2005-04-11 10:45
Number of contained objects............: 5
```

```
Job container state...................: waiting
.....................................3 - jobs waiting
.....................................0 - jobs active
.....................................2 - jobs ok
.....................................0 - jobs failed

Job name..............................: tstbase 1.0:inst
..Overall state.......................: WAITING
.....................................2 - job targets waiting
.....................................0 - job targets active
.....................................0 - job targets ok
.....................................0 - job targets failed
..Job order number....................: 0
..Created at..........................: 2005-06-30 13:00
..Activate at.........................: 2005-07-05 01:00
..Status message......................: OK
..Target..............................: KKKKK01B
....Task..............................: install
....Job state.........................: WAITING
....Task..............................: install
....Started at........................: 2005-07-05 01:00
....Completed at......................: 2005-04-11 10:45
....Created at........................: 2005-04-11 10:45
....Data sent in per cent.............:
....Status message....................: OK
....Ordered by........................: KKKKK01
....Type..............................: mandatory
....Comment...........................:
..Target..............................: KKKKK01C
....Task..............................: install
....Job state.........................: WAITING
....Task..............................: install
....Started at........................: 2005-07-05 01:00
....Completed at......................: 2005-04-11 10:45
....Created at........................: 2005-04-11 10:45
....Data sent in per cent.............:
```

```
....Status message.....................: OK
....Ordered by........................: KKKKK01
....Type..............................: mandatory
....Comment...........................:

Job name..............................: DMS Editor 6.01.0000:Local Install
..Overall state.......................: WAITING
.....................................2 - job targets waiting
.....................................0 - job targets active
.....................................0 - job targets ok
.....................................0 - job targets failed
..Job order number....................: 1
..Created at..........................: 2005-06-30 13:00
..Activate at.........................: 2005-07-05 01:00
..Status message......................: OK
..Target..............................: KKKKK01B
....Task..............................: install
....Job state.........................: WAITING
....Task..............................: install
....Started at........................: 2005-07-05 01:00
....Completed at......................: 2005-04-11 10:45
....Created at........................: 2005-04-11 10:45
....Data sent in per cent.............:
....Status message.....................: OK
....Ordered by........................: KKKKK01
....Type..............................: mandatory
....Comment...........................:
..Target..............................: KKKKK01C
....Task..............................: install
....Job state.........................: WAITING
....Task..............................: install
....Started at........................: 2005-07-05 01:00
....Completed at......................: 2005-04-11 10:45
....Created at........................: 2005-04-11 10:45
....Data sent in per cent.............:
....Status message.....................: OK
....Ordered by........................: KKKKK01
....Type..............................: mandatory
....Comment...........................:
```

```
Job name.............................: tstbase 1.0:remdir
..Overall state......................: unknown (9)
.....................................0 - job targets waiting
.....................................0 - job targets active
.....................................0 - job targets ok
.....................................0 - job targets failed
..Job order number...................: 2
..Created at.........................: 2005-06-30 13:00
..Activate at........................: 2005-07-05 01:00
..Status message.....................: Job setup warning. No target instal
.....................................+ lations found in evaluation step.

Job name.............................: tstbase 1.0:setdir
..Overall state......................: unknown (9)
.....................................0 - job targets waiting
.....................................0 - job targets active
.....................................0 - job targets ok
.....................................0 - job targets failed
..Job order number...................: 3
..Created at.........................: 2005-06-30 13:00
..Activate at........................: 2005-07-05 01:00
..Status message.....................: Job setup warning. No target instal
.....................................+ lations found in evaluation step.

Job name.............................: tstkpk01 1.0:inst
..Overall state......................: WAITING
.....................................2 - job targets waiting
.....................................0 - job targets active
.....................................0 - job targets ok
.....................................0 - job targets failed
..Job order number...................: 4
..Created at.........................: 2005-06-30 13:00
..Activate at........................: 2005-07-05 01:00
..Status message.....................: OK
..Target.............................: KKKKK01B
....Task.............................: install
....Job state........................: WAITING
....Task.............................: install
....Started at.......................: 2005-07-05 01:00
....Completed at.....................: 2005-04-11 10:45
....Created at.......................: 2005-04-11 10:45
....Data sent in per cent............:
```

```
....Status message.....................: OK
....Ordered by........................: KKKKK01
....Type..............................: mandatory
....Comment...........................:
..Target..............................: KKKKK01C
....Task..............................: install
....Job state.........................: WAITING
....Task..............................: install
....Started at........................: 2005-07-05 01:00
....Completed at......................: 2005-04-11 10:45
....Created at........................: 2005-04-11 10:45
....Data sent in per cent.............:
....Status message.....................: OK
....Ordered by........................: KKKKK01
....Type..............................: mandatory
....Comment...........................:


SDCMD<A000000>: OK
```

The following command allows you to see the attributes of a special job of this container:

```
cadsmcmd jobContainer action=showAttrJobs name=" q_grp [04/11/02
10:44:10.093]"jobName=" tstbase 1.0:inst"

CA IT Client Manager r12

ITCM Command Line Version 12.8.0.xxxx

Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "wwwww01b" as user "<default user>" ...OK.

-------------------------------------------------------------------------------
List of attributes of job "q_grp [04/11/02 10:44:10.093]" from container "tstbase
1.0:inst"
-------------------------------------------------------------------------------
Job name.............................: tstbase 1.0:inst
Software.............................: tstbase 1.0
Procedure............................: inst
Deliver at...........................: 2005-06-30 13:00
Activate at..........................: 2005-07-05 01:00
Status message.......................: OK
Overall state........................: WAITING
.....................................2 - job targets waiting
.....................................0 - job targets active
.....................................0 - job targets ok
.....................................0 - job targets failed
Job task.............................: install
Pre action...........................: logoff user
Post action..........................: no action
Prompt user..........................: No
Characteristic flags.................:
.....................................prevent user logon during job execution
.....................................local time/exact
.....................................calendar controlled
.....................................resolve query groups
Delivery calendar....................:
User parameters......................:
Prompt during........................: 1 Days, 0 Hours
Job timeout..........................: 7 Days, 0 Hours

SDCMD<A000000>: OK
```

createGroup—Create Group

This action allows you to create a software group or a procedure group.

This action has the following format:

```
swLibrary action=createGroup name=group_name
    [superGroup=super_group_name]
    [GroupType={SWG|PG}]
    [comment=comment]
```

name

Specifies the name of the group to be created.

superGroup

Specifies the name of the super group where the new group will be created as a subgroup. The super group must be a software group. If the parameter is not coded, the group will be created on top level, that is in folder "Software Package Library".

GroupType

Specifies the type of group to be created. The following values are valid:

SWG

Defines the group as a software group. This is the default.

PG

Defines the group as a procedure group.

comment

Comment on the groups.

Example:

Suppose a software group named "swg_1" should be created and after this, a procedure group named "pg_1," which is a subgroup of the already existing software group "swg_1". The following commands create these groups.

```
cadsmcmd swlibrary action=createGroup name=swg_1

cadsmcmd swlibrary action=createGroup name=pg_1 groupType=PG superGroup=swg_1
```

To check the creation the following command will list all the software and procedure groups available:

```
cadsmcmd swlibrary action=listGroups recursive



CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
List of groups in the software library
--------------------------------------------------------------------------------
Software Groups
..Group name.........................: Catalog
....Comment..........................: CA-Software Delivery Reserved Group
....Creation date....................: 2005-04-09 11:26
....Change date......................: 2005-04-09 11:26
..Group name.........................: Software Delivery
....Comment..........................: CA-Software Delivery Reserved Group
....Creation date....................: 2005-04-09 11:26
....Change date......................: 2005-04-09 11:27
..Group name.........................: swg_1
```

```
....Comment.............................:
....Creation date.......................: 2005-04-10 08:51
....Change date.........................: 2003-04-10 09:11
....Procedure Groups
......Group name........................: pg_1
........Comment.........................:
........Creation date...................: 2005-04-10 09:11
........Change date....................: 2005-04-10 09:11

SDCMD<A000000>: OK
```

Adding the argument "recursive" is important, otherwise the procedure group "pg_1" as a subgroup will not be listed, but only the top level groups.

```
cadsmcmd swlibrary action=listGroups


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "kkkkk01b" as user "<default user>" ...OK.

--------------------------------------------------------------------------------
List of groups in the software library
--------------------------------------------------------------------------------
Software Groups
..Group name............................: Catalog
....Comment.............................: CA-Software Delivery Reserved Group
....Creation date.......................: 2005-04-09 11:26
....Change date.........................: 2005-04-09 11:26
..Group name............................: Software Delivery
....Comment.............................: CA-Software Delivery Reserved Group
....Creation date.......................: 2005-4-09 11:26
....Change date.........................: 200504-09 11:27
..Group name............................: swg_1
....Comment.............................:
```

```
....Creation date.....................: 2005-04-10 08:51
....Change date.......................: 2005-04-10 09:11

SDCMD<A000000>: OK
```

Now assume that a software group "swg_2" should also be created and a software group "swg_2_1" that is a subgroup of swg_2. Furthermore, another procedure group named pg_2 is needed. The following commands will generate these groups and the intended hierarchy:

```
cadsmcmd swlibrary action=createGroup name=swg_2 groupType=SWG

cadsmcmd swlibrary action=createGroup name=swg_2_1 superGroup=swg_2

cadsmcmd swlibrary action=createGroup name=pg_2 groupType=PG
```

Now the recursive list of groups shows the following:

```
cadsmcmd swlibrary action=listGroups recursive


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "kkkkk01b" as user "<default user>" ...OK.

------------------------------------------------------------------------------
List of groups in the software library
------------------------------------------------------------------------------
Software Groups
..Group name..........................: Catalog
....Comment...........................: CA-Software Delivery Reserved Group
....Creation date.....................: 2005-04-09 11:26
....Change date.......................: 2005-04-09 11:26
..Group name..........................: Software Delivery
....Comment...........................: CA-Software Delivery Reserved Group
....Creation date.....................: 2005-04-09 11:26
....Change date.......................: 2005-04-09 11:27
..Group name..........................: swg_1
....Comment...........................:
....Creation date.....................: 2005-04-10 08:51
....Change date.......................: 2005-04-10 09:11
....Procedure Groups
```

```
......Group name.....................: pg_1
........Comment......................:
........Creation date.................: 2005-04-10 09:11
........Change date..................: 2005-04-10 09:11
..Group name.........................: swg_2
....Comment..........................:
....Creation date....................: 2005-04-10 09:28
....Change date......................: 200504-10 09:28
....Software Groups
......Group name.....................: swg_2_1
........Comment......................:
........Creation date.................: 2005-04-10 09:28
........Change date..................: 2005-04-10 09:28
Procedure Groups
..Group name.........................: pg_2
....Comment..........................:
....Creation date....................: 2005-04-10 09:29
....Change date......................: 2005-04-10 09:29

SDCMD<A000000>: OK
```

deleteGroup—Delete Software or Procedure Group

This action allows you to delete a software or procedure group.

This action has the following format:

```
swLibrary action=deleteGroup name=groupname [recursive]
```

name

Specifies the name of the group to be deleted.

Group names not valid for this method are "All Software" and "Catalog".

recursive

Not only the group will be deleted, but also all recursively contained subgroups.

Products or procedures assigned to a group to be deleted will be unlinked from that group but not deleted.

Note: If the group to be deleted does not exist, the action will be carried out with success (return code: SDCMD<A000000>: OK).

Example:

Suppose the software group "swg_2" is no longer needed and should be deleted. Also, all subgroups of this group should be deleted. The following command will execute the deletion.

```
cadsmcmd swlibrary action=deleteGroup name=swg_2 recursive
```

Launching

```
cadsmcmd swlibrary action=deleteGroup name=swg_2
```

will only delete the group specified. This will fail if the group swg_2 contains a subgroup and swg_2 is the only supergroup for this subgroup.

If there are other supergroups the subgroup will be unlinked from the specified group and the group will be deleted.

deliverGroupToStage—Deliver Items of Software Group To Scalability Servers

This action allows you to deliver the items of a software group to scalability servers.

This action has the following format:

```
swLibrary action=deliverGroupToStage name=group_name
    {{computer=computer_name
    | {compgrp=computer_group_name | ccompgrp=(computer_group_name sep
    groupScope)}}
    [sep=separator_sign]
    [dname=distribution_name]
    cname=[name]
    [unsealed]
    [{nolinkage|transaction[rollback]|synchronized}[nocascade]]
    [deliveryTime="YYYY-MM-DD hh:mm"]
    [calendarname=deliverycalendar]
    [globalTime]
    [{{area=area_name} | {domain=domain_name} | toAllAreas}]
    [sendTime="YYYY-MM-DD hh:mm"]
    [haltTime="YYYY-MM-DD hh:mm"]
    [jcPriority={1,...,10}]
    [distPriority={1,...,10}]
```

area

Specifies the name of a CA ITCM domain the distribution will be sent to.

The parameter can be coded more than once to address a list of CA ITCM domains.

If this parameter is coded, the distribution is sealed and sent to its destinations.

If neither the area nor the domain nor the toAllAreas parameter is coded, the distribution remains unsealed and can further be manipulated.

This parameter should not be coded with the domain or toAllAreas parameters.

(enterprise manager only)

calendarName

Specifies the name of the calendar that controls the time when the evaluation can take place.

ccompgrp

Specifies the computer group to be addressed as a pair of group name and scope. By default, name and scope are separated by a dot ("."), but an alternate separator can be specified with the "sep" option. The separator should **not** be used as part of the names.

For the group scope only the following values are valid.

local

The group is created at the local domain manager,

global

The group has been created at the enterprise manager and may be replicated.

The "ccompgrp" can be coded with the "compgrp" in one call.

cname

Specifies the the optional unique name of a job container.

If no cname parameter is given, a name is generated internally and the command line interface returns this identifier.

compgrp

Specifies the name of a server group, which contains the scalability servers to be addressed.

The parameter can be coded more than once to address more than one server group.

The compgrp and computer parameter should not be mixed (domain manager only).

computer

Specifies the name of the scalability server to be addressed.

The parameter can be coded more than once to address more than one scalability server.

The compgrp and computer parameter should not be mixed.

(Domain manager only)

deliveryTime

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

distPriority

Specifies the priority that will be assigned to the distribution.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded priority 5 is assumed.

dname

Specifies the name of the distribution.

If not specified the name is automatically generated.

(Enterprise manager only)

domain

Specifies the name of a CA ITCM domain group the distribution will be sent to.

The parameter can be coded more than once to specify a list of CA ITCM domain groups.

If this parameter is coded the distribution is sealed and sent to its destinations.

If neither the area nor the domain nor the toAllAreas parameter is coded, the distribution remains unsealed and can further be manipulated.

This parameter should not be coded with the area or toAllAreas parameters.

(enterprise manager only)

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

haltTime

Specifies the date and time at which a distribution order should be halted. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm".

(Enterprise manager only)

jcPriority

Specifies the priority of the job container.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded the priority is not changed.

name

Specifies the name of a software group of which the members will be delivered to the specified scalability server.

noCascade

The job linkage option is set to "Ignore cascading".

This parameter is allowed only if "transaction" or "synchronized" are coded.

noLinkage

Run the job independently of the other jobs in the container.

rollback

Set the job linkage option to "Enable Transaction".

sendTime

Specifies the time to send the associated distribution to the addressed domain managers. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm". If the parameter is not coded, the actual date and time is used.

(Enterprise manager only)

sep

Specifies an alternate separator used with the "ccompgroup" option. The separator should not be part of the group name code with the "ccompgroup" options.

synchronized

The job linkage option is set to "Synchronized job execution".

transaction

The job linkage option is set to "Batch job execution".

This is the default.

toAllAreas

If coded the related distribution will be sent to all CA ITCM domains registered at the enterprise manager.

If this parameter is coded then the distribution will be sealed and automatically sent to its destinations after the command has been successfully completed.

If neither the area nor the domain nor the toAllAreas parameter is coded, the distribution remains unsealed and can further be manipulated.

(Enterprise manager only)

unsealed

The addressed job container remains unsealed after the generated job has successfully been created.

(Domain manager only)

If neither area, nor domain, nor toAllAreas is coded, the distribution container remains unsealed and further manipulation is possible.

Example:

Suppose the items of the software group named "swg_1" should be delivered to all scalability server servers. The command is launched at a domain manager. The swg_1 shows the following contents:

```
cadsmcmd swlibrary action=listSWG name=swg_1


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM
```

```
------------------------------------------------------------------------------
List members of software group "swg_1"
------------------------------------------------------------------------------
tstbase 1.0 <Generic>, <Regular>, 22622(0), CA, http://supportconnect.ca.com/
tstkpk01 1.0 <Generic>, <Regular>, 22752(0), CA, http://supportconnect.ca.com/
tstkpk02 1.0 <Generic>, <Regular>, 22832(0), CA, http://supportconnect.ca.com/
DMS Editor 6.01.0000 <MSI>, <Regular>, 22972(0), CA, http://supportconnect.ca.com/
pg_1 <Procedure group>
```

Note: The entries of the subgroup pg_1 have been ignored because procedure groups are not valid for delivering products to scalability servers.

```
SDCMD<A000000>: OK

cadsmcmd swlibrary action=listPG name=pg_1

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "wwwww01b" as user "<default user>" ...OK.

------------------------------------------------------------------------------
List members of procedure group "pg_1"
------------------------------------------------------------------------------
remdir (tstbase 1.0 <1>)
setdir (tstbase 1.0 <2>)

SDCMD<A000000>: OK
```

The following command will provide the delivery:

```
cadsmcmd swlibrary action=deliverGroupToStage name=swg_1 compgrp="SD scalability
servers"

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "wwwww01b" as user "<default user>" ...OK.

A container with the name SD scalability servers [04/10/02 14:26:39.578] was created.
Evaluate group swg_1
Evaluate group swg_1/pg_1
Generate job "tstbase 1.0:delivery proc" : OK.
Generate job "tstkpk01 1.0:delivery proc" : OK.
Generate job "tstkpk02 1.0:delivery proc" : OK.
Generate job "DMS Editor 6.01.0000:delivery proc" : OK.

SDCMD<A000000>: OK
```

The contents of the generated job container is shown by the following command:

```
cadsmcmd jobContainer action=listJobs name="SD staging servers [04/10/02
14:26:39.578]"

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "wwwww01b" as user "<default user>" ...OK.

------------------------------------------------------------------------------
Job list of job container "SD scalability servers [04/10/02 14:26:39.578]"
------------------------------------------------------------------------------
Job name.............................: tstbase 1.0:delivery proc
..Task...............................: deliver
..Overall state......................: WAITING
..Job order number...................: 1

Job name.............................: tstkpk01 1.0:delivery proc
..Task...............................: deliver
..Overall state......................: WAITING
..Job order number...................: 2

Job name.............................: tstkpk02 1.0:delivery proc
..Task...............................: deliver
..Overall state......................: WAITING
..Job order number...................: 3

Job name.............................: DMS Editor 6.01.0000:delivery proc
..Task...............................: deliver
..Overall state......................: WAITING
..Job order number...................: 4

------------------------------------------------------------------------------
Number of jobs read: 4
Number of jobs listed: 4
SDCMD<A000000>: OK
```

listGroups—List Groups

This action lists all software and procedure groups.

This action has the following format:

```
swLibrary action=listGroups [recursive]
```

recursive

Lists subdirectories recursively.

If this parameter is omitted, only the top-level groups are listed.

Example

```
cadsmcmd swlibrary action=listgroups recursive
```

The following information is listed:

```
--------------------------------------------------------------------------------
List of goups in the software library
--------------------------------------------------------------------------------
Software Groups
..Group name.........................: Catalog
....Comment..........................: CA-Software Delivery Reserved Group
....Creation date....................: 2001-07-05 10:28
....Change date......................: 2001-08-28 17:42
..Group name.........................: Software Delivery
....Comment..........................: CA-Software Delivery Reserved Group
....Creation date....................: 2001-07-05 10:29
....Change date......................: 2001-09-06 10:36
..Group name.........................: swg 0
....Comment..........................:
....Creation date....................: 2001-09-12 10:35
....Change date......................: 2001-09-12 10:35
..Group name.........................: swg 1
....Comment..........................: Software Group 4 test
....Creation date....................: 2001-09-12 08:54
....Change date......................: 2001-09-12 10:35
....Software Groups
......Group name.....................: swg 0
........Comment......................:
........Creation date.................: 2001-09-12 10:35
........Change date..................: 2001-09-12 10:35
......Group name.....................: swg 2
........Comment......................: embedded Software Group of level 1
........Creation date.................: 2001-09-12 09:19
........Change date..................: 2001-09-12 09:31
```

```
........Software Groups
..........Group name...................: swg 3
............Comment...................: embedded Software Group of level 2
............Creation date..............: 2001-09-12 09:31
............Change date................: 2001-09-12 10:16
............Software Groups
..............Group name..............: swg 4
................Comment................: I'm an embedded Software Group too
................Creation date..........: 2001-09-12 10:15
................Change date............: 2001-09-12 10:15
............Procedure Groups
..............Group name..............: pg 2
................Comment................: I'm an embedded procedure group
................Creation date..........: 2001-09-12 10:16
................Change date............: 2001-09-12 10:16
....Procedure Groups
......Group name.......................: pg 0
........Comment........................:
........Creation date...................: 2001-09-12 10:35
........Change date....................: 2001-09-12 10:35
Procedure Groups
..Group name...........................: pg 0
....Comment............................:
....Creation date......................: 2001-09-12 10:35
....Change date.......................: 2001-09-12 10:35
..Group name...........................: pg 1
....Comment............................: procedure group 4 test
....Creation date......................: 2001-09-12 08:54
....Change date.......................: 2001-09-12 08:54

SDCMD<A000000>: OK
```

listMembership—List Membership

This action lists the memberships of a group; in other words, it shows those groups where it is assigned as a subgroup.

This action has the following format:

`swLibrary action=listMembership name=`*`group_name`*

name

Specifies the name of a software or procedure group of which the memberships will be listed.

Example:

To get information about those software groups containing a procedure group pg_1 enter:

```
cadsmcmd swlibrary action=listMembership name=pg_1
```

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
"pg_1" is member of the following Groups
--------------------------------------------------------------------------------
swg_1 (Software group, 2005-04-10 08:51)
swg_2.1 (Software group, 2005-04-10 09:28)

SDCMD<A000000>: OK
```

modifyGroup—Modify Attributes of Software or Procedure Group

This action modifies attributes of a software group or a procedure group.

This action has the following format:

```
swLibrary action=modifyGroup name=group_name
    [newName=new_group_name]
    [comment=comment]
```

comment

Specifies a new comment on the group.

Note: To erase an existing comment, enter the empty string ("").

name

Specifies the name of a computer group to be modified.

newName

Specifies the new name of the computer group.

new name of a computer group.

Example:

Suppose the Software Group named "swg_2_1" should be modified. The new name of the group is "swg_2.1" and a comment "4 test only" is added. The following command will provide the modifications.

```
cadsmcmd swlibrary action=modifyGroup name=swg_2_1 newName=swg_2.1 comment="4 test only"
```

undeliverGroupFromStage—Undeliver Items of Software Group From scalability servers

This action "undelivers" the items of a software group from scalability servers, in other words, they will be removed from the software library of the scalability server.

This action has the following format:

```
swLibrary action=undeliverGroupFromStage name=group_name
    {{computer=computer_name
    | {compgrp=computer_group_name | ccompgrp=(computer_group_name sep
    groupScope)}}
    [sep=separator_sign]
    [dname=distribution_name]
    cname=[name]
    [unsealed]
    [{nolinkage|transaction[rollback]|synchronized}[nocascade]]
    [deliveryTime="YYYY-MM-DD hh:mm"]
    [calendarname=deliverycalendar]
    [globalTime]
    [{{area=area_name} | {domain=domain_name} | toAllAreas}]
    [sendTime="YYYY-MM-DD hh:mm"]
    [haltTime="YYYY-MM-DD hh:mm"]
    [jcPriority={1,...,10}]
    [distPriority={1,...,10}]
```

area

Specifies the name of a CA ITCM domainthe distribution will be sent to.

The parameter can be coded more than once to address a list of CA ITCM domains.

If this parameter is coded then the distribution will be sealed and automatically sent to its destinations after the command has been successfully completed.

If neither the area nor the domain nor the toAllAreas parameter is coded, the distribution remains unsealed and can further be manipulated.

This parameter should not be coded with the domain or toAllAreas parameters.

(Enterprise manager only)

calendarName

Specifies the name of the calendar that controls the time when the evaluation can take place.

ccompgrp

Specifies the computer group to be addressed as a pair of group name and scope. By default, name and scope are separated by a dot ("."), but an alternate separator can be specified with the "sep" option. The separator should **not** be used as part of the names.

For the group scope only the following values are valid.

local

The group is created at the local domain manager,

global

The group has been created at the enterprise manager and may be replicated.

The "ccompgrp" can be coded with the "compgrp" in one call.

cname

Specifies an the optional unique name of a job container.

If no cname parameter is given, a name is generated internally and the command line interface returns this identifier.

compgrp

Specifies the name of the server group, which contains the scalability servers to be addressed.

The parameter can be coded more than once to address more than one server group.

The parameter should not be coded with the computer parameter.

computer

Specifies the name of the scalability server to be addressed.

The parameter can be coded more than once to address more than one scalability server.

The compgrp and computer parameter should not be mixed.

(Domain manager only)

deliveryTime

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

distPriority

Specifies the priority that will be assigned to the distribution.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded priority 5 is assumed.

dname

Specifies the name of the distribution

(Enterprise manager only)

domain

Specifies the name of a CA ITCM domain group the distribution will be sent to.

The parameter can be coded more than once to address a list of CA ITCM domain groups.

If this parameter is coded then the distribution will be sealed and automatically sent to its destinations after the command has been successfully completed.

If neither the area nor the domain nor the toAllAreas parameter is coded, the distribution remains unsealed and can further be manipulated.

This parameter should not be coded with the area or toAllAreas parameters.

(Enterprise manager only)

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

haltTime

Specifies the date and time at which a distribution order should be halted. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm".

jcPriority

Specifies the priority of the job container.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded priority 5 is assumed.

name

Specifies the name of a software group of which the members will be undelivered from the specified scalability server.

noCascade

The job linkage option is set to "Ignore cascading".

This parameter is allowed only if "transaction" or "synchronized" are coded.

noLinkage

Run the job independently of the other jobs in the container.

rollback

Set the job linkage option to "Enable Transaction".

sendTime

Specifies the time to send the associated distribution to the addressed domain managers. The date has to be specified in the ISO format "YYYY-MM-DD hh:mm". If the parameter is not coded, the actual date and time is used.

(for enterprise manager only)

sep

Specifies an alternate separator used with the "ccompgroup" option. The separator should not be part of the group name code with the "ccompgroup" options.

synchronized

The job linkage option is set to "Synchronized job execution".

toAllAreas

If coded the related distribution will be sent to all CA ITCM domains registered at the enterprise manager.

If this parameter is coded then the distribution will be sealed and automatically sent to its destinations after the command has been successfully completed.

If neither the area nor the domain nor the toAllAreas parameter is coded, the distribution remains unsealed and can further be manipulated.

"toAllAreas" should not be coded with the "area" or "domain" parameters.

(Enterprise manager only)

transaction

The job linkage option is set to "Batch job execution".

unsealed

The addressed job container remains unsealed after the generated job has successfully been created.

(Domain manager only)

If neither area, nor domain, nor toAllAreas is coded, the distribution container remains unsealed and further manipulation is possible.

Software Group Management

This section contains the following topics:

addGroupToSWG—Add Group To Software Group

This action allows you to add new subgroups to a software group.

This action has the following format:

```
swLibrary action=addGroupToSWG [name=group_name]
    {subGroup=group_name}
```

name

Specifies the name of a software group that will be expanded by the new subgroup of the parameter subGroup.

If this parameter is not coded, then the top-level folder "Software Library" is assumed.

subGroup

Specifies the name of a subgroup to be added to the group.

The subgroup must already exist as a group in the software library.

The parameter can be coded more than once to add a list of subgroups.

addItemToSWG—Add Item To Software Group

This action allows you to add products to a software group.

This action has the following format:

```
swLibrary action=addItemToSWG
    name=software_group_name
    {item=item_name version=item_version
    | {itemList=(item_name/item_version)}}
```

item

Specifies the name of the item to be added.

If this parameter is coded, the parameter version becomes mandatory.

This parameter should not be coded with the itemList parameter.

itemList

Specifies an item, identified by the pair (itemname itemversion) to be added to the software group.

This parameter could be coded more than once to create a list of items to be added.

Because the slash ("/") is used as a delimiter, the item name or item version containing slashes should be enclosed in quotes (").

This parameter should not be coded with the item parameter.

name

Specifies the name of a software group to which the item will be added.

version

Specifies the version of the item to be added.

This parameter should be coded with the item parameter only.

Notes: There are two possibilities to code an item to be added: The first is by using the parameters item and version, the other by using the parameter itemList. Both formats should not be mixed in the command, but at least one of them should be used.

The items to be added must already be defined in the software library at the time of command invocation. This command does not create any new item in the software library.

Using the itemList parameter at the batch or at the command-line interface and having an item name or an item version containing slashes you have to duplicate the slashes in the item name or version.

If the itemList parameter contains any blanks enclose it in quotes ("). (This holds for command line or batch usage).

listSWG—List Products of Software Group

This action lists the products of a software group recursively.

This action has the following format:

```
swLibrary action=listSWG name=groupname [recursive]
```

name

Specifies the name of a software group to be listed.

recursive

Lists subdirectories recursively.

If this parameter is omitted, only the top-level groups are listed.

Example

Consider a software group swg_1. Before modifying, the group, the structure of the group, and the contents should be listed. This action is done using the following commands:

```
cadsmcmd swlibrary action=listSWG name=swg_1

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.
Trace mode: FILE
Trace file name: C:\ProgramFiles\CA\DSM\SD\..\logs\cadsmcmd.log
Connecting to manager "<default manager>" as user "<default user>" ...OK.
Manager:mymanager
Domain:mydomain
Domain type:Domain
Supporting: CO CCNF USD OSIM AM
--------------------------------------------------------------------------------
List members of software group "swg_1"
--------------------------------------------------------------------------------
tstbase 1.0 <Generic>, <Regular>, 22622(0), CA, http://supportconnect.ca.com/
tstkpk01 1.0 <Generic>, <Regular>, 22752(0), CA, http://supportconnect.ca.com/
tstkpk02 1.0 <Generic>, <Regular>, 22832(0), CA, http://supportconnect.ca.com/
tstkpk03 1.0 <Generic>, <Regular>, 22962(0), CA, http://supportconnect.ca.com/
pg_1 <Procedure group>
SDCMD<A000000>: OK

cadsmcmd swlibrary action=listPG name=pg_1

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "kkkkk01b" as user "<default user>" ...OK.

--------------------------------------------------------------------------------
List members of procedure group "pg_1"
--------------------------------------------------------------------------------
inst (tstbase 1.0 <1>)
inst (tstkpk03 1.0 <2>)
SDCMD<A000000>: OK
```

The entry of tstkpk03 is obsolete and should be removed. To perform this action, use the following command:

```
cadsmcmd swlibrary action=removeItemFromSWG name=swg_1 item=tstkpk03 version=1.0
```

The removal does not mean that the item tstkpk03 is deleted from the software library. It is still available at "All Software" and at all other groups where it has been assigned. It is just removed from the group swg_1.

If, besides tstkpk03, the entry tstbase should also be removed, then the following command lets you remove both entries by one call:

```
cadsmcmd swlibrary action=removeItemFromSWG name=swg_1 itemList=(tstbase/1.0)
itemList=("tstkpk03" /"1.0")
```

Now consider a product named "DMS Editor" of Version 6.01.0000 that should be added to the group. The following command performs the assignment:

```
cadsmcmd swlibrary action=addItemToSWG name=swg_1 item="DMS Editor"
version=6.01.0000
```

If another product should be added (for example, "DMS Interpreter" of Version 6.01.0000), this action can be done by the following command in one call:

```
cadsmcmd swlibrary action=addItemToSWG name=swg_1 itemList=("DMS Editor" /6.01.0000)
itemList=("DMS Interpreter" /6.01.0000)
```

The group now has the following contents:

```
cadsmcmd swlibrary action=listSWG name=swg_1
```

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.
Connecting to manager "kkkkk01b" as user "<default user>" ...OK.
--------------------------------------------------------------------------------
List members of software group "swg_1"
--------------------------------------------------------------------------------
tstbase 1.0 <Generic>, <Regular>, 22622(0), CA, http://supportconnect.ca.com/
tstkpk01 1.0 <Generic>, <Regular>, 22752(0), CA, http://supportconnect.ca.com/
tstkpk02 1.0 <Generic>, <Regular>, 22832(0), CA, http://supportconnect.ca.com/
DMS Editor 6.01.0000 <MSI>, <Regular>, 22972(0), CA http://supportconnect.ca.com/
pg_1 <Procedure group>
SDCMD<A000000>: OK
```

```
cadsmcmd swlibrary action=listPG name=pg_1

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.
Connecting to manager "kkkkk01b" as user "<default user>" ...OK.
--------------------------------------------------------------------------------
List members of procedure group "pg_1"
--------------------------------------------------------------------------------
inst (tstbase 1.0 <1>)
inst (tstkpk03 1.0 <2>)
SDCMD<A000000>: OK
```

Consider the following situation:

```
cadsmcmd swlibrary action=listGroups recursive

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.
Connecting to manager "kkkkk01b" as user "<default user>" ...OK.
--------------------------------------------------------------------------------
List of groups in the software library
--------------------------------------------------------------------------------
Software Groups
..Group name.........................: Catalog
....Comment..........................: CA-Software Delivery Reserved Group
....Creation date....................: 2003-04-09 11:26
....Change date......................: 2003-04-09 11:26
..Group name.........................: Software Delivery
....Comment..........................: CA-Software Delivery Reserved Group
....Creation date....................: 2003-04-09 11:26
....Change date......................: 2003-04-09 11:27
..Group name.........................: swg_1
....Comment..........................:
....Creation date....................: 2003-04-10 08:51
....Change date......................: 2003-04-10 10:33
....Procedure Groups
......Group name.....................: pg_1
........Comment......................:
........Creation date.................: 2003-04-10 09:11
........Change date..................: 2003-04-10 10:13
..Group name.........................: swg_2
....Comment..........................:
....Creation date....................: 2003-04-10 09:28
....Change date......................: 2003-04-10 09:28
```

```
....Software Groups
......Group name.....................: swg_2.1
........Comment......................: 4 test only
........Creation date................: 2003-04-10 09:28
........Change date..................: 2003-04-10 09:57
..Group name.........................: swg_2.1
....Comment..........................: 4 test only
....Creation date....................: 2003-04-10 09:28
....Change date......................: 2003-04-10 09:57
Procedure Groups
..Group name.........................: pg_2
....Comment..........................:
....Creation date....................: 2003-04-10 09:29
....Change date......................: 2003-04-10 09:29
SDCMD<A000000>: OK
```

The software group swg_2.1 should be removed as a subgroup from group swg_2, and pg_2 should be added as a new subgroup of swg_2. The following commands rearrange the hierarchy:

```
cadsmcmd swlibrary action=removeGroupFromSWG name=swg_2 subgroup=swg_2.1

cadsmcmd swlibrary action=addGroupToSWG name=swg_2 subgroup=pg_2
```

Now the hierarchy is as follows:

```
cadsmcmd swlibrary action=listGroups recursive

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.
Connecting to manager "kkkkk01b" as user "<default user>" ...OK.
--------------------------------------------------------------------------------
List of groups in the software library
--------------------------------------------------------------------------------
Software Groups
..Group name.........................: Catalog
....Comment..........................: CA-Software Delivery Reserved Group
....Creation date....................: 2003-04-09 11:26
....Change date......................: 2003-04-09 11:26
..Group name.........................: Software Delivery
....Comment..........................: CA-Software Delivery Reserved Group
....Creation date....................: 2003-04-09 11:26
....Change date......................: 2003-04-09 11:27
```

```
..Group name.........................: swg_1
....Comment..........................:
....Creation date....................: 2003-04-10 08:51
....Change date......................: 2003-04-10 10:58
....Procedure Groups
......Group name.....................: pg_1
........Comment......................:
........Creation date.................: 2003-04-10 09:11
........Change date..................: 2003-04-10 10:13
..Group name.........................: swg_2
....Comment..........................:
....Creation date....................: 2003-04-10 09:28
....Change date......................: 2003-04-10 11:11
....Procedure Groups
......Group name.....................: pg_2
........Comment......................:
........Creation date.................: 2003-04-10 09:29
........Change date..................: 2003-04-10 09:29
..Group name.........................: swg_2.1
....Comment..........................: 4 test only
....Creation date....................: 2003-04-10 09:28
....Change date......................: 2003-04-10 09:57
Procedure Groups
..Group name.........................: pg_2
....Comment..........................:
....Creation date....................: 2003-04-10 09:29
....Change date......................: 2003-04-10 09:29
SDCMD<A000000>: OK
```

If swg_2.1 is not assigned somewhere outside swg_2 (in other words, swg_2 is its only supergroup), the command to remove this subgroup from swg_2 fails with an error A001563.

removeGroupFromSWG—Remove Group from Software Group

This action removes subgroups from a software group. This does not mean that the subgroups will be deleted.

This action has the following format:

```
swLibrary action=removeGroupFromSWG [name=group_name]
    {subGroup=group_name}
```

name

Specifies the name of the group from which the subgroup is removed.

If this parameter is not coded the top level folder "Software Package Library" is assumed.

subGroup

Specifies the name of a subgroup to be removed from the group.

The parameter can be coded more than once to remove a list of subgroups.

The subgroups are unlinked from the group.

This process fails if the specified group is the only supergroup of the subgroup. Such a subgroup can only be deleted with the action=deleteGroup.

Removing a subgroup from a group does not mean that the subgroup is deleted.

removeItemFromSWG—Remove Item from Software Group

This action removes products from a software group.

This action has the following format:

```
swLibrary action=removeItemFromSWG
    name=software_group_name
    {item=item_name
    version=item_version |
    {itemList=(item_name/item_version)}}
```

item

Specifies the name of the item to be removed.

itemList

Specifies an item, identified by the pair (itemname itemversion), to be removed from the software group.

This parameter could be coded more than once to create a list of items to be removed.

Because the slash ("/") is used as a delimiter, the item name or item version containing slashes should be enclosed in quotes (").

name

Specifies the name of a software group from which the item will be removed.

version

Specifies the version of the item to be removed.

Notes: There are two possibilities to code an item to be removed. The first is by using the parameters item and version, the other by using the parameter itemList. Both formats should not be mixed in the command, but at least one of them should be used.

Removing an item from a software group does not mean that the item is deleted from the software library. Other assignments to other software groups remain unchanged. Using the itemList parameter at the batch or at the command-line interface and if an item name or an item version contains any slashes you have to duplicate the slashes in the item name or version.

If the itemList parameter contains any blanks enclose it in quotes ("). (This holds for command line or batch usage).

Examples (see definition on page 311)

Catalog Group Management

Catalog handling is a special form of Software Group Handling. The following catalog methods are provided.

They are valid on domain manager only.

This section contains the following topics:

createCG—Create Catalog Group

This action allows you to create a catalog group.

This action is valid only for domain managers

This action has the following format:

```
swLibrary action=createCG
    name=computer_group_name
    [groupScope={global | local}]
```

groupScope

Specifies the scope of the group specified. The following values are valid for this option:

Global

The group is a global group, created at an enterprise manager. It may be replicated to a domain manager.

Local

Specifies a local group, created at the domain manager the CADSMCMD is connected to.

If "groupScope" is not coded then the CADSMCMD first tries to locate the group as a local group. If this fails then it is tried to locate the group as a global one.

name

Specifies the name of a computer group to create a catalog group.

Note: For an example of the use of the createCG command, see the example for the listCG command.

listCG—List Catalog Group

This action allows you to list all existing catalog groups.

This action is valid for domain managers only.

This action has the following format:

```
swLibrary action=listCG [detailed]
```

detailed

Lists the assigned products.

Example:

To see all catalog groups defined, enter the following command:

```
cadsmcmd swlibrary action=listCG
```

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.
Trace mode: FILE
Trace file name: C:\ProgramFiles\CA\DSM\SD\..\logs\cadsmcmd.log
```

```
Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM
--------------------------------------------------------------------------------
List of catalog groups
--------------------------------------------------------------------------------
c_grp (cgrp, local)
q_grp (qgrp, local)
t_grp (tgrp, local)
SDCMD<A000000>: OK
```

If the products assigned to the catalog groups should also be listed, then the following command should be launched:

```
cadsmcmd swlibrary action=listCG detailed
```

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.
Trace mode: FILE
Trace file name:C:\ProgramFiles\CA\DSM\SD\..\logs\cadsmcmd.log
Connecting to manager "default manager" as user "<default user>" ...OK.
Manager:mymanager
Domain:mydomain
Domain type:Domain
Supporting:CO CCNF USD OSIM
--------------------------------------------------------------------------------
List of catalog groups
--------------------------------------------------------------------------------
c_grp (cgrp, local)
..DMS Editor 6.01.0000 <MSI>, <Regular>, 22422<0>, CA, http://supportconnect.ca.com/
q_grp (qgrp, local)
..DM Script Generator 4.0 <Generic>, <Regular>, 22722<0>, CA,
http://supportconnect.ca.com/
t_grp (tgrp, local)
..SD Boot Server Extension for WinNT/W2K 4.0 <Generic>, <Regular>, 22922<0>, CA,
http://supportconnect.ca.com/
SDCMD<A000000>: OK
```

Now assume the catalog group "t_grp" has become obsolete and therefore should be removed. In addition, a new catalog group should be added for the computer group tq_grp. The following commands provide this reorganization:

```
cadsmcmd swlibrary action=removeCG name=t_grp
```

```
cadsmcmd swlibrary action=createCG name=tq_grp
```

To add the item "Data Transport Service for Win32" of Version "2.0 SP1 ENU" to the catalog group tq_grp the following command can be entered:

```
cadsmcmd swlibrary action=addItemToSWG name=tq_grp item="Data Transport Service for
Win32" version="2.0 SP1 ENU"
```

You can remove items from a catalog group using the action removeItemFromSWG.

The catalog list now goes as follows:

```
cadsmcmd swlibrary action=listCG detailed

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.
Trace mode: FILE
Trace file name: C:\ProgramFiles\CA\DSM\SD\..\logs\cadsmcmd.log
Connecting to manager "default manager" as user "<default user>" ...OK.
Manager:mymanager
Domain:mydomain
Domain type:Domain
Supporting:CO CCNF USD OSIM

--------------------------------------------------------------------------------
List of catalog groups
--------------------------------------------------------------------------------
c_grp (cgrp, local)
..DMS Editor 6.01.0000 <MSI>, <Regular>, 22422<0>, CA, http://supportconnect.ca.com/
q_grp (qgrp, local)
..DM Script Generator 4.0 <Generic>, <Regular>, 22722<0>, CA,
http://supportconnect.ca.com/
tq_grp (tgrp, local)
..Data Transport Service for Win32 2.0 SP1 ENU <Generic>, <Regular>, 22922<0>, CA,
http://supportconnect.ca.com/
SDCMD<A000000>: OK
```

removeCG—Remove Catalog Group

This action removes a catalog group.

This action is valid only for domain managers

This action has the following format:

```
swLibrary action=removeCG name=catalog_group_name
```

name

Specifies the name of the catalog group to be deleted.

Note: For an example of the use of the removeCG command, see the example for the listCG command.

Procedure Group Management

This section contains the following topics:

addProcToPG—Add Procedures To Procedure Group

This action allows you to add procedures of products to a procedure group.

The item must already be defined in the software library and the procedure must already be defined for the item.

This action has the following format:

```
swLibrary action=addProcToPG name=procedure_group_name
    procedure=procedure_name
    item=item_name
    version=item_version
```

name

Specifies the name of the procedure group to which the procedure will be added.

procedure

Specifies the name of a procedure which will be added.

item

Specifies the name of the item from where the procedure is taken.

version

Specifies the version of the item from where the procedure is taken.

listPG—List Procedures of Procedure Group

This action lists the procedures assigned to a procedure group.

This action has the following format:

```
swLibrary action=listPG name=procedure_group_name
```

name

Specifies the name of the procedure group to be listed.

The following information will be shown:

```
--------------------------------------------------------------------------------
List members of procedure group "pg 0"
--------------------------------------------------------------------------------
Reinstall Package (xxbl0del (SXP) 2.0/00 <1>)
Reinstall Package (xxbl1del (SXP) 2.0/00 <2>)
Reinstall Package (xxbl4del (SXP) 2.0/00 <3>)

SDCMD<A000000>: OK
```

The number in pointed brackets shows the order number of the procedure in the procedure group.

Example:

To list the contents of the procedure group pg_1 the following command can be entered:

```
cadsmcmd swlibrary action=listPG name=pg_1


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off
```

```
Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
List members of procedure group "pg_1"
--------------------------------------------------------------------------------
inst (tstbase 1.0 <1>)
inst (tstkpk03 1.0 <2>)

SDCMD<A000000>: OK
```

The tstbase entry should be replaced by two other entries invoking the procedures remdir and setdir. The following commands will modify the contents as required:

```
cadsmcmd swlibrary action=removeProcFromPG name=pg_1 procedure=inst item=tstbase
version=1.0

cadsmcmd swlibrary action=addProcToPG name=pg_1 procedure=remdir item=tstbase
version=1.0

cadsmcmd swlibrary action=addProcToPG name=pg_1 procedure=setdir item=tstbase
version=1.0
```

The group pg_1 now shows the following contents:

```
cadsmcmd swlibrary action=listPG name=pg_1

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "kkkkk01b" as user "<default user>" ...OK.

--------------------------------------------------------------------------------
List members of procedure group "pg_1"
--------------------------------------------------------------------------------
inst (tstkpk03 1.0 <1>)
remdir (tstbase 1.0 <2>)
setdir (tstbase 1.0 <3>)

SDCMD<A000000>: OK
```

The procedures of a procedure group are executed at the target systems in the sequence they were ordered in the procedure group. This order number of a procedure is shown in pointed brackets. To execute the inst procedure in a correc manner the remdir and setdir must run first. The following command reorders the procedure group:

```
cadsmcmd swlibrary action=repositionItem name=pg_1 procedure=inst item=tstkpk03
version=1.0 position=3
```

Now the group pg_1 shows the following contents:

```
cadsmcmd swlibrary action=listPG name=pg_1

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "kkkkk01b" as user "<default user>" ...OK.

------------------------------------------------------------------------------
List members of procedure group "pg_1"
------------------------------------------------------------------------------
remdir (tstbase 1.0 <1>)
setdir (tstbase 1.0 <2>)
inst (tstkpk03 1.0 <3>)

SDCMD<A000000>: OK
```

removeProcFromPG—Remove Procedures from Procedure Group

This action removes procedures from a procedure group.

This action has the following format:

```
swLibrary action=removeProcFromPG name=procedure_group_name
    procedure=procedure_name
    item=item_name
    version=item_version
```

item

Specifies the name of the item where the procedure is located.

name

Specifies the name of the procedure group from which the procedure will be removed.

procedure

Specifies the name of a procedure to be removed.

version

Specifies the version of the item where the procedure is located.

Note: This function only unlinks the procedure from the procedure group. It does not remove the procedure from the item nor does it affect any other procedure and software groups as long as they do not include the procedure group in question.

repositionItem—Reposition Item in Procedure Group

This action reorders a procedure group by assigning a new order number to an element of the procedure group.

This action has the following format:

```
swlibrary action=repositionItem name=procedure_group_name
    procedure=procedure_name
    item=item_name
    version=item_version
    position=new_position
```

item

Specifies the name of the item where the procedure is located.

name

Specifies the name of the procedure group to be reordered.

position

Specifies the new position of the procedure.

The position has to be greater than zero and should not exceed the number of procedures in the procedure group.

procedure

Specifies the name of a procedure which will be repositioned.

version

Specifies the version of the item where the procedure is located.

Product Management

This section contains the following topics:

archive—Archive Product in software library

This action will archive the specified product at the specified location.

This action has the following format:

```
swlibrary action=archive
    item=item_name
    version=item_version
    path=path_name
    [purge]
    [onserver]
    [comment=comment]
```

comment

Optional parameter: a comment for the action.

item

Specifies the name of the product to be archived.

onserver

If coded, the destination is settled on the server, otherwise not.

path

Specifies the name of an empty directory where the product is to be archived.

purge

When the product is already archived, the product files will be deleted from the software library.

version

Specifies the version of the product to be archived.

export—Export Product

This action will export the specified product to the specified location.

This action has the following format:

```
swlibrary action=export
    item=item_name
    version=item_version
    path=path_name
    [onserver]
    [as_image]
```

as_image

If coded the software will be exported to a library image that can be used by the Software Management Installer to grab packages from a CD.

item

Specifies the name of the product to be exported.

version

Specifies the version of the product to be exported.

path

Specifies the destination where the exported product is stored. The directory must be empty.

onserver

If it is coded, the destination is settled on the manager, otherwise not.

list—List Products in Software Library

This action displays the list of products from the software library. An additional filter can be specified to restrict the list to just those products that fulfill the filter criteria. If no filters are given, the complete list of products is shown. The filter is limited to100 bytes.

This action has the following format:

```
swlibrary action=list [{filter="filter" | filterfile=filename }]
```

filter

Specifies the expression to confine the amount of information listed, in this case, the products to be shown.

You can also use the attributes shown by this list action.

Note: You can display a list of virtual or regular software packages by specifying one of the following filters:

filter="Package format=Virtual"

filter="Package format=Regular"

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command.

Example:

To display the list of products from the software library, enter the following command:

```
cadsmcmd swlibrary action=list
```

The following information is shown:

```
--------------------------------------------------------------------------------
List of library items
--------------------------------------------------------------------------------
Item name........: tstool_1
..Item version........: 1.0
..Item type........: SOFTWARE
..Package type.....:Generic
..Package format...:Regular
..Package size.....:90
..Based on item........:
..Based on version........:
..Creation date........: 2009-09-25
..Creation time........: 11:39
..Change date........: 2009-09-25
..Change time........: 11:41
..Checksum enabled...:Yes
.........................
Item name........: tstool_1
..Item version........: 1.1
..Item type........: SOFTWARE
..Package type.....:Generic
..Package format...:Regular
..Package size.....:70
..Based on item........: tstool_1
..Based on version........: 1.0
..Creation date........: 2009-09-25
..Creation time........: 11:41
..Change date........: 2009-09-25
..Change time........: 11:45
..Checksum enabled...:Yes
--------------------------------------------------------------------------------
Total number of Items: 74
Total number of Items Read: 74
Total number of items shown: 74
SDCMD<A000000>: OK
```

To display the list of products from the software library starting with name "tst" and having version 1.1 using filters, enter the following command:

```
cadsmcmd swlibrary action=list filter="(Item name=tst* && Item version=1.1)"
```

The following information is shown:

```
-------------------------------------------------------------------------------
List of library items
-------------------------------------------------------------------------------
Item name........: tstool_1
..Item version........: 1.1
..Item type........: SOFTWARE
..Package type.....:Generic
..Package format...:Regular
..Package size.....:80
..Based on item........: tstool_1
..Based on version........: 1.0
..Creation date........: 2009-09-25
..Creation time........: 11:41
..Change date........: 2009-09-25
..Change time........: 11:45
..Checksum enabled...:Yes
-------------------------------------------------------------------------------
Total number of Items: 74
Total number of Items Read: 74
Total number of items shown:1
SDCMD<A000000>: OK
```

ListFailedJobs—List Failed Jobs

This action displays the list of failed jobs.

This action is valid only for domain managers

This action has the following format:

```
swlibrary action=listfailedjobs
    item=item_name
    version=version_name
    {alt |
    procedure=procedure_name
    [{filter=filter|filterfile=file_name}]}
```

alt

An alternate list is provided based on the item name and version only.

In case of coding this parameter the value of the item and version parameter might contain database supported wild cards, for example,

"%"

for an arbitrary, even empty sequence of characters

"_"

for an arbitrary single character

For Microsoft SQL Server also the square brackets ("[","]") have a special meaning. This has to be taken into account when coding the name.

filter

Specifies the expression to confine the amount of information listed, in this case the amount of products.

You can also use the attributes shown by this list action.

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command.

item

Specifies the name of the item for which the list is provided.

procedure

Specifies the name of the item procedure used in the jobs in question.

version

Specifies the version of the item for which the list is provided.

Example:

```
C:\>cadsmcmd local mydomain swlibrary action=listfailedjobs item=tstprod version=1.5
procedure=inst



CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM

------------------------------------------------------------------------------
List of jobs "tstprod 1.5:inst"
------------------------------------------------------------------------------
Job name........: tstprod 1.5:inst
..Job identifier........: 19438
..Job state........: ERROR
..Job order number........: 0
..Job task........: 1 - install
..Creation date........: 2005-02-14
..Creation time........: 11:13
..Delivery date........: 2005-02-14
..Delivery time........: 11:13
..Activation date........: 2005-02-14
..Activation time........: 11:13
..Targetsystem........: ZZZZ01B
....Computer job identifier........: 19470
....State........: EXECUTION_ERROR
```

```
....Error message........: Exit code 2 indicates possible error
....Error cause........: 228001
....Completion date........: 2005-02-14
....Completion time........: 11:14
Job name........: tstprod 1.5:inst
..Job identifier........: 19446
..Job state........: ERROR
..Job order number........: 2
..Job task........: 1 - install
..Creation date........: 2005-03-17
..Creation time........: 12:23
..Delivery date........: 2005-03-17
..Delivery time........: 12:23
..Activation date........: 2005-03-17
..Activation time........: 12:23
..Targetsystem........: ZZZZ01A
....Computer job identifier........: 19485
....State........: EXECUTION_ERROR
....Error message........: Exit code 2 indicates possible error
....Error cause........: 228001
....Completion date........: 2005-03-17
....Completion time........: 12:25


SDCMD<A000000>: OK
```

To list the failed jobs with Job order number as 0, enter the following command:

```
cadsmcmd swlibrary action=listfailedjobs item=tstprod version=1.5 procedure=inst
filter=”Job order number=0”
```

The list returned is as follows:

```
--------------------------------------------------------------------------------
List of jobs "tstprod 1.5:inst"
--------------------------------------------------------------------------------
Job name........: tstprod 1.5:inst
..Job identifier........: 19438
..Job state........: ERROR
..Job order number........: 0
..Job task........: 1 - install
..Creation date........: 2005-02-14
..Creation time........: 11:13
..Delivery date........: 2005-02-14
..Delivery time........: 11:13
..Activation date........: 2005-02-14
..Activation time........: 11:13
..Targetsystem........: ZZZZ01B
....Computer job identifier........: 19470
....State........: EXECUTION_ERROR
....Error message........: Exit code 2 indicates possible error
....Error cause........: 228001
....Completion date........: 2005-02-14
....Completion time........: 11:14

SDCMD<A000000>: OK
```

listJobs—List Jobs

This action lists the jobs related to the item is shown. The information also shows the computers addressed by the job and individual results of the job for each computer.

This action is valid only for domain managers

This action has the following format:

```
swlibrary action=listjobs
    item=item_name
    version=version_name
    {alt |
    procedure=procedure_name
    [{filter=filter|filterfile=file_name}]}
```

alt

An alternate list is provided based on the item name and version only.

An alternate list is provided based on the item name and version only.

In case of coding this parameter the value of the item and version parameter might contain database supported wild cards, for example,

"%"

for an arbitrary, even empty sequence of characters

"_"

for an arbitrary single character

For Microsoft SQL Server also the square brackets ("[","]") have a special meaning. This has to be taken into account when coding the name.

filter

Specifies the expression to confine the amount of information listed.

You can also use the attributes shown by this list action.

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command.

item

Specifies the name of the item for which the list is provided.

procedure

Specifies the name of the item procedure used in the jobs in question.

version

Specifies the version of the item for which the list is provided.

Example:

The following information will be shown:

```
C:\>cadsmcmd local mydomain swlibrary action=listjobs item=tstprod version=1.5
procedure=inst


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
List of jobs "tstprod 1.5:inst"
--------------------------------------------------------------------------------
Job name........: tstprod 1.5:inst
..Job identifier........: 19438
..Job state........: ERROR
..Job order number........: 0
..Job task........: 1 - install
..Creation date........: 2006-02-14
..Creation time........: 11:13
..Delivery date........: 2006-02-14
..Delivery time........: 11:13
..Activation date........: 2006-02-14
..Activation time........: 11:13
..Targetsystem........: ZZZZZ01A
....Computer job identifier........: 19472
....State........: EXECUTION_OK
....Error message........: OK
....Error cause........: 0
....Completion date........: 2006-02-14
....Completion time........: 11:18
..Targetsystem........: ZZZZZ01B
....Computer job identifier........: 19470
....State........: EXECUTION_ERROR
....Error
```

```
....Error message........: Exit code 2 indicates possible error
....Error cause.........: 228001
....Completion date.....: 2006-02-14
....Completion time.....: 11:14
........
Job name........: tstprod 1.5:inst
..Job identifier........: 19454
..Job state.........: OK
..Job order number........: 0
..Job task.........: 1 - install
..Creation date.....: 2006-02-14
..Creation time.....: 11:14
..Delivery date.....: 2006-02-14
..Delivery time.....: 11:13
..Activation date.....: 2006-02-14
..Activation time.....: 11:14
..Targetsystem.....: YYYYY
....Computer job identifier.....: 19481
....State.....: EXECUTION_OK
....Error message.....: OK
....Error cause.....: 0
....Completion date.....: 2006-02-14
....Completion time.....: 11:18
..Targetsystem.....: XXXX01
....Computer job identifier.....: 19479

....State.....: EXECUTION_OK
....Error message.....: OK
....Error cause.....: 0
....Completion date.....: 2006-02-14
....Completion time.....: 11:14

SDCMD<A000000>: OK
```

To list jobs that are in ERROR state, enter following command.

```
cadsmcmd swlibrary action=listjobs item=tstprod version=1.5 procedure=inst
filter="Job state=ERROR"
```

The list returned is as follows:

```
--------------------------------------------------------------------------------
List of jobs "tstprod 1.5:inst"
--------------------------------------------------------------------------------

Job name........: tstprod 1.5:inst
..Job identifier........: 19438
..Job state........: ERROR
..Job order number........: 0
..Job task........: 1 - install
..Creation date........: 2006-02-14
..Creation time........: 11:13
..Delivery date........: 2006-02-14
..Delivery time........: 11:13
..Activation date........: 2006-02-14
..Activation time........: 11:13
..Targetsystem........: ZZZZZ01A
....Computer job identifier........: 19472
....State........: EXECUTION_OK
....Error message........: OK
....Error cause........: 0
....Completion date........: 2006-02-14
....Completion time........: 11:18
..Targetsystem........: ZZZZZ01B
....Computer job identifier........: 19470
....State........: EXECUTION_ERROR
....Error
....Error message........: Exit code 2 indicates possible error
....Error cause........: 228001
....Completion date........: 2006-02-14
....Completion time........: 11:14

SDCMD<A000000>: OK
```

restore—Restore Product from Archive

This action restores the specified product from an archive in the software library.

This action has the following format:

```
swlibrary action=restore
    {item=item_name version=item_version
    | path=path_name }
```

item

Specifies the name of the product to be restored from archive.

path

Specifies the destination where the product to be restored has been archived.

version

Specifies the version of the product to be restored from archive.

Note: If the product is identified by name and version the archive is determined by the archive path information stored with the product. If the path name is given, the archived product is restored from there, and identified by the information also stored there.

sealItem—Seal Product

This action seals the specified product.

This action has the following format:

```
swlibrary action=sealitem
    item=item_name
    version=item_version
```

item

Specifies the name of the product to be sealed.

version

Specifies the version of the product to be sealed.

swpolicy—Software Policy Commands

This section contains the following topics:

addItem—Add Jobs to Unsealed Local Software Policy

This command adds jobs to an unsealed local software policy. The jobs to be added can be specified either by procedure or software groups, by a file containing all necessary information, or by directly passing the information for a job with the call. In case of a successful addition, the policy can be optionally sealed with this action.

This command has the following format:

```
swPolicy action=addItem
    name=software_policy_name
    {group=name_of_a_software_or_procedure_group|
    procedures=job_file|
    item=item_name
    version=item_version
    [task={activate | configure | install | uninstall}]
    procedure=procedure_name
    [installedWith=install_procedure_name]
    [after={exacttime | boottime}]
    [reinstall[={y|n}]]
    [preaction={none | reboot | logoff}]
    [postaction={none | reboot | logoff | rebootAtEnd | logoffAtEnd | shutdownAtEnd}]
    [promptUser[={y|n}]]
    [allowCancel[={y|n}]]
    [execTimedOut[={y|n}]]
    [offline[={y|n}]]
    [runAtShutdown[={y|n}]]
    [preventLogon[={y|n}]]
    [triggeredByScalSrv[={y|n}]]
    [globaltime[={y|n}]]
    [calendarname=delivery_calendar_name]
    [noCalendar[={y|n}]]
    [resolveQuery[={y|n}]]
    [stagingServer[={y|n}]]
    [parameters=user_parameters]
    [prompt=d.h]
    [jobTimeout=d.h]
    [repeat[={y|n}]]
    [seal[={y|n}]]
    [userJobMessage]
```

name

The name of the software policy the jobs will be added to.

group

This option specifies the name of a software or procedure group. The members of the group are used to add jobs to the software policy.

- The group specified is a “software group”:

  The jobs of the policy are generated from the default installation procedures of the group members, if there are any, or from the first installation procedure in order. The job options are taken from the defaults or are inherited from the procedure. If there are no install procedures for a member then the member is skipped and a warning is reported.

- The group specified is a “procedure group”:

  The jobs for the policy are generated from the procedures of the group. The job options are taken from the defaults or are inherited from the procedure.

If this option is not coded no groups are evaluated for populating the created software policy.

The group option should not be coded with the procedures or item option.

procedures

This option specifies the name of a file that contains the procedures for generating the jobs for the software policy. For each job to be created and assigned to the policy an entry of the following syntax has to be created in the file:

```
item=item_name version=item_version
procedure=procedure_name
task={activate | configure | install | uninstall}
[installedWith=install_procedure_name]
[after={exacttime | boottime}]
[reinstall[={y|n}]]
[preaction={none | reboot | logoff}]
[postaction={none | reboot | logoff | rebootAtEnd | logoffAtEnd | shutdownAtEnd}]
[promptUser[={y|n}]]
[allowCancel[={y|n}]]
[execTimedOut[={y|n}]]
[offline[={y|n}]]
[runAtShutdown[={y|n}]]
[preventLogon[={y|n}]]
[triggeredByScalSrv[={y|n}]]
[globaltime[={y|n}]]
[calendarname=delivery_calendar_name]
[noCalendar[={y|n}]]
[resolveQuery[={y|n}]]
[stagingServer[={y|n}]]
[parameters=user_parameters]
[prompt=d.h]
[jobTimeout=d.h]
[repeat[={y|n}]]
```

There should be only one entry per line and an entry should not span over more than one line. The entry has to begin with the parameters “item”, “version”, “procedure” and “task” and they have to appear in this sequence. The order of the remaining parameters is arbitrary. Parameter values containing blanks should be enclosed in quotes (“). If a parameter value contains a quote is should be duplicated.

For the meaning of the parameters please see the option "descriptions" below.

The procedures option should not be coded with the "group" or "item" option.

item

Specifies the name of a registered item product. The job will be generated for this item and added to the software policy.

The item option should not be coded with the "group" or "procedures" option.

version

Specifies the version of the item specified that is used for the job generation.

task

Specifies the type of procedure for which the job will be added to the software policy. The following task types are valid:

activate

The procedure selected is an activation procedure.

configure

The procedure selected is a configuration procedure.

install

The procedure selected is an installation procedure.

uninstall

The procedure selected is a un-installation procedure.

If the option is not coded then "install" is default.

procedure

Name of the item procedure that is used for generating the job to be added to the software policy

installedWith

If specified then it is the name of an installation procedure of the item specified. The job to be generated is only executed on those machines where the item is installed by this specified procedure. This parameter is ignored if the job to be added is an installation job itself.

If the option is not coded then the way the item is installed is not checked.

after

This option specifies when the job execution will start at the target systems. The following values are valid:

exacttime

The procedure starts processing as soon as the order has fallen due.

boottime

After falling due the processing of the procedure is deferred until the next (re-)boot time.

The default is “exacttime”.

reinstall

If “reinstall” or “reinstall=y” is coded then the item will be reinstalled at the targets each time the software policy evaluates. If “reinstall=n” is coded or the option is missing then the item will only be installed if it is missing at the targets. The option is valid for install procedures; for any other type of procedure this option is ignored.

preaction={none | reboot | logoff}]

Specifies any actions that must take place before the related job is processed at the target computers. The valid pre-actions are:

none

Performs no pre-actions.

reboot

Restarts the system before executing the job.

logoff

Logs off the users before executing the job.

If this option is not coded, the default is given by the item procedure’s options.

postaction={none | reboot | logoff | rebootAtEnd | logoffAtEnd | shutdownAtEnd}]

Specifies any actions that must take place after the related job is processed at the target computers. The following post-actions are valid:

none

Performs no post-actions.

reboot

Restarts the system after executing the job.

logoff

Logs off the users after executing the job.

rebootAtEnd

Specifies that the job requires a reboot after it has been processed. But the reboot can be deferred until all other jobs resulting from the same evaluation of the software policy have been processed or a subsequent job requires an immediate reboot before or after its processing.

logoffAtEnd

Specifies that the job requires a user to log off after it has been processed. But the logoff can be deferred until all other jobs resulting from the same evaluation of the software policy have been processed or a subsequent job requires an immediate restart or logoff before or after its processing.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

If this option is not coded, the default is given by the item procedure's options.

promptUser

If "promptUser" or "promptUser=y" has been specified then the user at the targets is prompted before the job is executed. If "prompt=n" is specified then the job is executed without any user prompting.

If this option is not coded the default is given by the item procedure's options.

allowCancel

If “allowCancel” or “allowCancel=y” is specified then the user at the targets can cancel the execution of the job when being prompted for execution. If “allowCancel=n” is specified then the user cannot cancel the job.

For enabling the cancel the prompting must be allowed. Otherwise an error is reported.

If this option is not coded the default is given by the item procedure’s options.

execTimedOut

If “execTimedOut” or “execTimedOut=y” is coded then the processing of the job is automatically started when the user prompting times out. If “execTimedOut=n” is coded then the job does not start at time out.

For enabling the start at time out the prompting must be allowed, otherwise an error is reported.

If this option is not coded the default is given by the item procedure’s options.

offline

If “offline” or “offline=y” is coded then connection between the target and the associated server will be released for the job’s execution; if “offline=n” is coded the connection is not released.

If this option is not coded the default is given by the item procedure’s options.

runAtShutdown

If “runAtShutdown” or “runAtShutdown=y” is coded then the job is only executed at system shutdown; if “runAtShutdown=n” is coded then the job is also executed at any other time it falls due.

If this option is not coded the default is given by the item procedure’s options.

This option applies to UNIX only.

preventLogon

If “preventLogon” or “preventLogon=y” is coded then a user logon attempt at the targets is rejected while the job is running. If a user is already logged on when the job starts then execution is deferred until the user logs off.

If “preventLogon=n” is coded then the processing of the job does not care whether a user is logged in or tries to login during run time.

If this option is not coded the default is given by the item procedure’s options.

This option applies to Windows targets only.

triggeredByScalSrv

If “triggeredByScalSrv” or “triggeredByScalSrv=y” is coded then the scalability servers associated with the targets initiate the job’s execution. If “triggeredByScalSrv=n” is coded the target initiates the execution at time of reboot or jobCheck.

Default is “triggeredByScalSrv”.

globaltime

If “globalTime” or “globalTime=y” is coded then all dates associated with the job are based on enterprise manager time, otherwise it is based on domain manager time.

Default is “globaltime=n”.

calendarname

Specifies the name of the calendar that is responsible for driving the delivery of the job to the scalability servers and targets.

noCalendar

If “noCalendar” or “noCalendar=y” is coded a possible execution calendar attached to the target computers will be ignored for the job processing. If “noCalendar=n” is coded then the calendar is taken into account.

Default is “noCalendar=n”.

resolveQuery

If “resolveQuery” or “resolveQuery=y” is coded and the group associated with the software template is a query group then this group is re-evaluated before the job is created. Otherwise no re-evaluation takes place.

Default is “resolveQuery=n”.

stagingserver

If “stagingserver” or “stagingserver=y” is coded the related item is staged at the scalability servers that are associated with addressed targets otherwise it is not staged. The option is ignored if the procedure is not an installation procedure. If it is not coded or “stagingserver=n” then the staging step is skipped.

Default is “stagingserver=n”.

parameters

Specifies the parameters that are passed to the procedure of the job. If an empty string is coded then it is assumed that there are no parameters.

If not coded then the user parameters are taken from the item procedure's options.

prompt

Specifies the time period in which the user will be re-prompted for the job execution start at the target when the user defers the execution. The period's format is "d.h" where "d" means days and "h" means hours and the range is 0.3 <= d.h <=7.0.

If the specified value is fallen below the allowed minimum then the minimum is set and if it exceeds the maximum then the maximum is set. No warning is given in these cases.

jobTimeout

Specifies the expiration date of the job in form of a time period. The period's format is "d.h" where "d" means days and "h" means hours. The range of the parameter is configurable at the manager's site.

If the specified value is fallen below the allowed minimum then the minimum is set and if it exceeds the maximum then the maximum is set. No warning is given in these cases. If not coded the default is given by 7.0.

repeat

If "repeat" or "repeat=y" is coded then the related job is repeated every time an evaluation takes place. If "repeat=n" is coded the related job is not repeated.

Default: is "repeat=n".

seal

If "seal" or "seal=y" is coded then the policy is sealed after the jobs have been added otherwise it remains unsealed for further modifications. The seal is only performed when the generating and adding of all required jobs has been successful and no errors have been reported for any jobs. Warnings for jobs generated do not suppress the seal.

Default is "seal=n".

userJobMessage

Specifies the custom administrator message for a job in a software policy.

Note. When the jobs added are given by a group or the procedures file then an error detected during the job generation is reported for the related job and the process continues with the generation of the next job. No rollback is performed.

create—Create a Local Software Policy

This command creates a local software policy based on the specified options. You can optionally link the software policy to a local or a global computer group and populate it with jobs during the creation phase. If you are able to link and populate the policy successfully, you can also seal it.

This command has the following format:

```
swPolicy action=create
    name=software_policy_name
    [target=computer_group_name
    [groupScope={local|global}]]
    [seal[={y|n}]]
    [{group=name_of_a_software_or_procedure_group|
    procedures=job_file}]
    [jcPriority=1…10]
    [{transaction|synchronized|nolinkage}]
    [rollback[={y|n}]]
    [cascade[={y|n}]]
    [{setup_jobs|setup&activate_jobs}]
    [regenerate[={y|n}]]
    [evaluateGroup[={y|n}]]
    [evaluationStart=YYYY-MM-DD hh:mm]
    [globalTime[={y|n}]]
    [evaluationPeriod=hours]
    [calendarname=calendar_name]
    [{adhoc_only|adhoc_always}]
    [comment=comment_on_software_policy]
    [userMessage]
```

name

The name of the software policy created is specified with this option. This name has to be unique.

target

The name specified with this option is the name of a computer group the software policy will be linked to. The computer group must already exist. It can either be a global group at the enterprise manager or a local or global group at the domain manager.

If this option is not coded then the software policy to be created is not linked to any computer group.

groupScope

Determines the type of computer group being linked to the software policy:

«not set»

If there is a local computer group of the specified name then this one is taken, otherwise it is looked for a global one.

local

The computer group has to be a local group.

global

The computer group has to be a global group.

seal

If “seal” or “seal=y” is coded then the created policy is sealed otherwise it remains unsealed for further modifications.

The seal option is only allowed when the policy is linked and populated with the creation and it is only processed if the link and population has been successful.

group

This option specifies the name of a software or procedure group. The members of the group are used to populate the software policy during the creation process with jobs.

- The group specified is a “software group”:

  The jobs of the policy are generated from the default installation procedures of the group members, if there are any, or from the first installation procedure in order. The job options are taken from the defaults or are inherited from the procedure.

- The group specified is a “procedure group”:

  The jobs for the policy are generated from the procedures of the group. The job options are taken from the defaults or are inherited from the procedure.

If this option is not coded no groups are evaluated for populating the created software policy.

The group option should not be coded with the procedures option.

procedures

This option specifies the name of a file that contains the information for generating the jobs for the software policy. For each job to be generated and assigned to the policy an entry of the following syntax has to be created in the file:

```
item=item_name version=item_version
procedure=procedure_name
task={activate | configure | install | uninstall}
[installedWith=install_procedure_name]
[after={exacttime | boottime}]
[reinstall={y|n}]
[preaction={none | reboot | logoff}]
[postaction={none | reboot | logoff | rebootAtEnd | logoffAtEnd | shutdownAtEnd}]
[promptUser={y|n}]
[allowCancel={y|n}]
[execTimedOut={y|n}]
[offline={y|n}]
[runAtShutdown={y|n}]
[preventLogon={y|n}]
[triggeredByScalSrv={y|n}]
[globaltime={y|n}]
[calendarname=delivery_calendar_name]
[noCalendar={y|n}]
[resolveQuery={y|n}]
[stagingServer={y|n}]
[parameters=user_parameters]
[prompt=d.h]
[jobTimeout=d.h]

[repeat={y|n}]
```

There should be only one entry per line and an entry should not span over more than one line. The entry has to begin with the parameters “item”, “version”, “procedure” and “task” and they have to appear in this sequence. The order of the remaining parameters is arbitrary. Parameter values containing blanks should be enclosed in quotes (“). If a parameter value contains a quote it should be duplicated.

For the meaning of the parameters please have a look at “cadsmcmd swPolicy action=addItem (see page 339)”.

The procedures option should not be coded with the group option.

jcPriority

This option specifies the job container priority used by the policy to create a job container for launching its SD orders. A valid priority is one of the values 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. 10 is the lowest priority and 1 is the highest. If the option is not coded the priority 5 is set.

{transaction|synchronized|nolinkage}

This option specifies the job linkage within the job container created by the policy for launching the SD orders.

«not coded»

No linkage between orders.

transaction

Batch job execution without transaction.

synchronized

Synchronized job execution.

nolinkage

No linkage between the orders.

rollback

If the option “rollback” or “rollback=y” is coded then the job linkage is set to "Enable Transaction". If the option is coded as “rollback=n” then “Enable Transaction” is not set.

Default is “rollback=n”.

The parameter should only be coded if parameter "transaction" is coded with this command.

cascade

If the option “cascade” or “cascade=y” is coded then the SD will resolve the dependencies of cascaded installation request and add the missing installation orders automatically. If the option is coded as “cascade=n” then the dependency is not resolved.

Default is “cascade”.

{setup_jobs|setup&activate_jobs}

This option drives the results of the policy’s evaluation:

«not coded»

The SD checks only for conformity while evaluating the policy and reports the non conform targets but no jobs are generated for transferring the non conformant targets into conformant.

setup_jobs

As a result of the evaluation the SD generates jobs for transferring all non conformant targets into conformant ones but these jobs are not launched for execution. The latter is left to the user.

setup&activate_jobs

As a result of the evaluation the SD generates jobs for transferring all non conformant targets into conformant ones and launches them for execution.

regenerate

If the option “regenerate” or “regenerate=y” is coded then all jobs that failed earlier and have not been completed successfully yet are rescheduled for execution for achieving conformity. If the option “regenerate=n” is coded then this rescheduling is skipped.

Default is “regenerate”.

evaluateGroup

If the option “evaluateGroup” or “evaluateGroup=y” is coded then in case of a query group associated with the policy this group will be re-evaluated during the policy’s evaluation too. If “evaluationGroup=n” is coded then this re-evaluation of the group will not happen.

Default is “evaluateGroup=n”.

evaluationStart

This option specifies the earliest point in time when the evaluation of the policy will start. The value is specified in ISO format “YYYY-MM-DD hh:mm” where YYYY specifies the year, MM the month, DD the current day of month, hh the hour and mm the minute. If this option is not coded the current time is taken.

globalTime

This option specifies whether the time specifications with the policy are based on enterprise time or domain time. If this option is coded then the time is enterprise base otherwise it is domain time.

evaluationPeriod

This option specifies the period the evaluations of the policy takes place. The value is specified in hours. If 0 is specified then the evaluation happens only once.

calendarname

This option specifies the name of the calendar used by the SD for scheduling the time driven actions of the policy.

{adhoc_only|adhoc_always}

This option specifies whether SD evaluates this policy when a target reports its inventory or not and it specifies the mode of processing:

«not coded»

The policy evaluation is not initiated by any target inventory reports.

adhoc_only

The evaluation is started by inventory reports of new or re-installed targets.

ad_hoc_always

The evaluation is started by any inventory reports from targets.

comment

The option specifies a comment that is stored with the policy.

userMessage

Specifies the custom administrator message for a software policy.

Note: You should be aware of the following:

- In case of an error detected during link, or seal phase further processing is stopped and the error is reported. No rollback takes place; the policy remains as it is at time of failure.
- In case of an error detected during populating the policy the error is reported for the related job but the populating continues, only a possibly required seal is suppressed.

removeItem—Remove Jobs from Unsealed Local Software Policy

This action removes jobs from an unsealed local software policy. The jobs to be removed are either directly passed with the command, or specified by a procedure or software group. In case of a successful removal the policy can optionally be sealed.

The syntax of the action is as follows:

```
swPolicyaction=removeItem
    name=«software_policy_name»
    {group=«name_of_a_software_or_procedure_group»|
    {jobNames=«job_name», ...}}
    [seal[={y|n}]]
```

name

The name of the software policy the specified jobs will be removed from.

group

This option specifies the name of a software or procedure group. The members of the group are used to determine the jobs to be removed from the software policy.

- The group specified is a “software group”:

  The jobs to be removed from the policy are determined by the members of the group and their related default installation procedures. If a member does not have a default installation group then the first installation procedure in order is taken. If there is a member not exposing any install parameters then the member is skipped and a warning is reported.

- The group specified is a “procedure group”:

  The jobs to be removed from the policy are determined by the procedure members of the group.

If a job to be removed is not assigned to the policy an error is reported for this job and the action continues with the next job.

If this option is not coded no groups are evaluated for removing jobs from the software policy.

The group option should not be coded with the "jobNames" option.

jobNames

Specifies the name of the jobs that will be removed from the software policy. The option can be coded more than once to pass a list of job names.

If a job is specified in the list that is not assigned to the policy then an error is reported and the action executes the next job from the list.

The jobNames option should not be coded with the "group" option.

seal

If “seal” or “seal=y” is coded then the policy is sealed after the jobs have been removed otherwise it remains unsealed for further modifications. The seal is only performed when the removal of all required jobs has been successful and no errors have been reported for any of these jobs. Warnings for jobs do not suppress the seal.

The seal is also suppressed if the policies job assignment falls empty by this action.

listItem—Show Information about Jobs Assigned to Software Policy

This action provides information about the jobs assigned to the specified software policy.

The syntax of the action is as follows:

```
swPolicyaction=listItem
    name=software_policy_name
    [swpScope={distributed | local}]
    [detailed]
```

name

The name of the software policy its jobs will be listed.

swpScope

This option determines the type of software policy involved in the action.

«not coded»

If there is a locally created software policy of the required name then this one is taken, if not it is looked for a policy distributed from an enterprise manager. The latter is on domain managers only.

distributed

The specified software policy has to be a policy created at the enterprise manager and distributed to the addressed manager.
(For domain managers only)

local

The specified software policy has to be a locally created policy.

Note: On enterprise manager all software policies have to be of type local.

detailed

If coded a detailed list is provided otherwise only the names of the assigned jobs are listed.

The short list provided is of the following format:

```
-----------------------------------------------------------------
List Jobs Assigned To Software Policy «software policy name»
Short
-----------------------------------------------------------------
«name of job 1»
         •
         •
         •
«name of job n»
```

The detailed list is of the following format.

```
-----------------------------------------------------------------
List Jobs Assigned To Software Policy «software policy name»
Detailed
-----------------------------------------------------------------
Job name: «name of job 1»
..Item name     : «name of item»
..Item version  : «version of item»
..Procedure name: «procedure name»
..Job task      : {configure|activate|
install|uninstall}
..Position      : «position of the job»
..Reinstall required    : {yes|no}
..Preaction     : {none|reboot|logoff}
..Postaction    : {none|reboot|logoff|
rebootAtEnd|logoffAtEnd}
..Prompt user   : {yes|no}
....Allow cancel: {yes|no}
....Execute jobs if prompt times out    : {yes|no}
....Prompt during (days/hours)   : d/h
..Run offline from server       : {yes|no}
..Run at shutdown       : {yes|no}
..Prevent user login    : {yes|no}
..Job startup   : {local|global} time/{exact|at boot}
..Delivery calendar     : «delivery calendar name»
..Ignore job calendars at target : {yes|no}
```

```
..Resolve query groups   : {yes|no}
..Stage item     : {yes|no}
..User parameters        : «user parameters»
..Job time out (days/hours)      : d/h
Job name: «name of job 2»
         •
         •
         •
Job name: «name of job n»
```

modifyItem—Modify Job of Unsealed Local Software Policy

This command modifies a job of an unsealed local software policy. You can optionally seal the policy with this command.

This command has the following format:

```
swPolicyaction=modifyItem
    name=software_policy_name
    jobname=name_of_job
    [newjobname=new_job_name]
    [after={exacttime | boottime}]
    [reinstall[={y|n}]]
    [preaction={none | reboot | logoff}]
    [postaction={none | reboot | logoff | rebootAtEnd | logoffAtEnd | shutdownAtEnd}]
    [promptUser[={y|n}]]
    [allowCancel[={y|n}]]
    [execTimedOut[={y|n}]]
    [offline[={y|n}]]
    [runAtShutdown[={y|n}]]
    [preventLogon[={y|n}]]
    [triggeredByScalSrv[={y|n}]]
    [globaltime[={y|n}]]
    [calendarname=delivery_calendar_name]
    [noCalendar[={y|n}]]
    [resolveQuery[={y|n}]]
    [stagingServer[={y|n}]]
    [parameters=user_parameters]
    [prompt=d.h]
    [jobTimeout=d.h]
    [repeat[={y|n}]]
    [seal[={y|n}]]
    [userJobMessage]
```

name

The name of the software policy that contains the job to be modified.

jobname

The name of the job that is modified.

newjobname

The new name of the job. If not coded the old name remains.

after

This option times when the job will start executing at the target systems. The following values are valid:

exacttime

The job starts processing as soon as the order has fallen due.

boottime

After falling due the processing of the job is deferred until the next (re-)boot time.

If not coded this option is not changed.

reinstall

If “reinstall” or “reinstall=y” is coded then the item will be reinstalled at the targets each time the software policy evaluates. If “reinstall=n” is coded then the item will only be installed if it is missing at the targets. The option is valid for install procedures only; for any other type of procedure this option is ignored.

If not coded this option is not changed.

preaction={none | reboot | logoff}]

Specifies any actions that must take place before the related job is processed at the target computers. The valid pre-actions are:

none

Performs no pre-actions.

reboot

Restarts the system before executing the job.

logoff

Logs off the users before executing the job.

If not coded, this option is not changed.

postaction={none | reboot | logoff | rebootAtEnd | logoffAtEnd | shutdownAtEnd}]

Specifies any actions that must take place after the related job has been processed at the target computers. The following post-actions are valid:

none

Performs no post-actions.

reboot

Restarts the system after executing the job.

logoff

Logs off the users after executing the job.

rebootAtEnd

Specifies that the job requires a reboot after it has been processed. But the reboot can be deferred until all other jobs resulting from the same evaluation of the software policy have been processed or a subsequent job requires an immediate reboot before or after being processed.

logoffAtEnd

Specifies that the job requires a user logoff after it has been processed. But the logoff can be deferred until all other jobs resulting from the same evaluation of the software policy have been processed or a subsequent job requires an immediate reboot or logoff before or after being processed.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

If not coded, this option is not changed.

promptUser

If “promptUser” or “promptUser=y” has been specified then the user at the targets is prompted before the job is executed. If “promptUser=n” is specified then the job is executed without any user prompting.

If not coded this option is not changed.

allowCancel

If “allowCancel” or “allowCancel=y” is specified then the user at the targets can cancel the execution of the job when being prompted for execution. If “allowCancel=n” is specified then the user cannot cancel the job.

For enabling the cancel the prompting must be required otherwise an error is reported.

If not coded this option is not changed.

execTimedOut

If “execTimedOut” or “execTimedOut=y” is coded then the processing of the job is automatically started when the user prompting times out. If “execTimedOut=n” is coded then the procedure does not start at time out.

For enabling the start at time out the prompting must be allowed, otherwise an error is reported.

If not coded this option is not changed.

offline

If “offline” or “offline=y” is coded then connection between the target and the associated server will be released for the job’s execution; if “offline=n” is coded the connection is not released.

If not coded this option is not changed.

runAtShutdown

If “runAtShutdown” or “runAtShutdown=y” is coded then the job is only executed at system shutdown; if “runAtShutdown=n” is coded then the job is also executed at any other time it falls due.

If not coded this option is not changed.

This option applies to UNIX only.

preventLogon

If “preventLogon” or “preventLogon=y” is coded then a user logon attempt at the targets is rejected while the job is running. If a user is already logged on when the job starts then execution is deferred until the user logs off.

If “preventLogon=n” is coded then the processing of the job does not care whether a user is logged in or tries to login during run time.

If not coded this option is not changed.

This option applies to Windows targets only.

triggeredByScalSrv

If “triggeredByScalSrv” or “triggeredByScalSrv=y” is coded then the scalability servers associated with the targets initiate the job’s execution. If “triggeredByScalSrv=n” is coded the target initiates the execution at time of reboot or jobCheck.

If not coded this option is not changed.

globaltime

If “globalTime” or “globalTime=y” is coded then all dates associated with the job are based on enterprise manager time, if “globaltime=n” is coded it is based on domain manager time.

If not coded this option is not changed.

calendarname

Specifies the name of the calendar that is responsible for driving the delivery of the job to the scalability servers and targets. If the empty string (“”) is coded then delivery calendar entry is emptied.

If not coded this option is not changed.

noCalendar

If “noCalendar” or “noCalendar=y” is coded a possible execution calendar attached to the target computers will be ignored for the job processing. If “noCalendar=n” is coded then the calendar is taken into account.

If not coded this option is not changed.

resolveQuery

If “resolveQuery” or “resolveQuery=y” is coded and the group associated with the software template is a query group then this group is re-evaluated before the job is created for the related procedure. If “resolveQuery=n” is coded then no re-evaluation takes place.

If not coded this option is not changed.

stagingserver

If “stagingserver” or “stagingserver=y” is coded the related item is staged at the scalability servers that are associated with the targets otherwise it is not staged. The option is ignored if the job is not an installation job. If “stagingserver=n” is coded then the staging step is skipped

If not coded this option is not changed.

parameters

Specifies the parameters that are passed to the job procedure. If an empty string (“”) is coded then it is assumed that no parameters have to be passed.

If not coded this option is not changed.

prompt

Specifies the time period in which the user will be re-prompted for the job procedure’s start at the target when the user defers the execution. The period’s format is “d.h” where "d" means days and "h" means hours and the range is 0.3 <= d.h <=7.0.

If the specified value is fallen below the allowed minimum then the minimum is set and if it exceeds the maximum then the maximum is set. No warning is given in these cases.

If not coded this option is not changed.

jobTimeout

Specifies the expiration date of the procedure in form of a time period. The period's format is "d.h" where d means days and h means hours. The range of the parameter is configurable at the manager's site.

If the specified value is fallen below the allowed minimum then the minimum is set and if it exceeds the maximum then the maximum is set. No warning is given in these cases.

If not coded this option is not changed.

repeat

If "repeat" or "repeat=y" is coded then the related job is repeated every time an evaluation takes place. If "repeat=n" is coded the related job is not repeated.

If not coded this option is not changed.

seal

If "seal" or "seal=y" is coded then the policy is sealed after the modification has been completed otherwise it remains unsealed for further modifications. The seal is only performed when the modifications have been successful and no error has been reported. Warnings do not suppress the seal.

If not coded, this option is not changed.

userJobMessage

Modifies the custom administrator message for a job in a software policy.

showAttrItem—Show Information about Job Assigned to Software Policy

This action provides information about the specified job assigned to the specified software policy.

The syntax of the action is as follows:

```
swPolicyaction=showAttrItem
    name=software_policy_name
    [swpScope={distributed | local}]
    jobname=job_name
```

name

The name of the software policy at which the job to be shown is located.

swpScope

This option determines the type of software policy involved in the action.

«not coded»

If there is a locally created software policy of the required name then this one is taken, if not it is looked for a policy distributed from an enterprise manager. The latter is on domain managers only.

distributed

The specified software policy has to be a policy created at the enterprise manager and distributed to the addressed manager.

local

The specified software policy has to be a locally created policy.

Note: On enterprise manager all software policies have to be of type local.

jobname

Name of the job to be listed.

The information listed is of the following format.

```
-----------------------------------------------------------------
Show Jobs «job name» of Software Policy «software policy name»
-----------------------------------------------------------------
Item name       : «name of item»
Item version    : «version of item»
Procedure name  : «procedure name»
Job task: {configure|activate|
install|uninstall}
Position: «position of the job»
Reinstall required      : {yes|no}
Preaction       : {none|reboot|logoff}
Postaction      : {none|reboot|logoff|
rebootAtEnd|logoffAtEnd}
Prompt user     : {yes|no}
..Allow cancel  : {yes|no}
..Execute jobs if prompt times out      : {yes|no}
..Prompt during (days/hours)    : d/h
Run offline from server : {yes|no}
Run at shutdown : {yes|no}
Prevent user login      : {yes|no}
Job startup     : {local|global} time/{exact|at boot}
Delivery calendar       : «delivery calendar name»
Ignore job calendars at target  : {yes|no}
Resolve query groups    : {yes|no}
Stage item      : {yes|no}
User parameters : «user parameters»
Job time out (days/hours)       : d/h
```

repositionItem—Reorder Jobs Assigned to Unsealed Local Software Policy

This action allows the reordering of the jobs assigned to the specified unsealed local software policy. This means that the sequence of the job execution at the targets is changed.

The syntax of the action is as follows:

```
swPolicyaction=repositionItem
    name=software_policy_name
    jobName=name_of_the_job
    position=new_position_of_the_job
    [seal[={y|n}]]
```

name

The name of the software policy that is reordered.

jobName

Name of the job that is re-positioned.

position

New position of the specified job in the software policy.

seal

If “seal” or “seal=y” is coded then the policy is sealed after the re-positioning has been completed otherwise it remains unsealed for further modifications. The seal is only performed when the re-positioning has been successful and no error has been reported. Warnings do not suppress the seal.

link—Link Software Policy to Computer Group

This action links or re-links a software policy to an existing computer group. The computer group can be local or global. The software policy has to be unsealed otherwise the action will abort with error. For re-linking a policy either the option “currentGroup” or the option “replace” has to be coded with the call otherwise the re-linking will fail.

The software policy has to be unsealed but of any scope. The latter also halts for the computer group to which the policy is linked.

With the link action the policy can optionally be sealed.

The syntax of the action is as follows:

```
swPolicyaction=link
    name=software_policy_name
    [swpScope={distributed | local}]
    group=computer_group_name
    [groupScope={global | local}]
    [{currentGroup=current_computer_group_name
    [currentGroupScope={global | local}]| replace}]
    [seal[={y|n}]]
```

name

The name of the software policy that is linked to the specified computer group.

swpScope

This option determines the type of software policy involved in the action.

«not coded»

If there is a locally created software policy of the required name then this one is taken, if not it is looked for a policy distributed from an enterprise manager. The latter is on domain managers only.

distributed

The specified software policy has to be a policy created at the enterprise manager and distributed to the addressed manager.

local

The specified software policy has to be a locally created policy.

Note: On enterprise manager all software policies have to be of type local.

group

This option specifies the name of an existing computer group to which the software policy specified will be assigned.

groupScope

This option specifies the type of group to which the software policy will be assigned.

global

The computer group has been created at the enterprise manager and maybe replicated to the domain.

local

The computer group has been created at the domain manager.

On enterprise managers all computer groups are global.

If the option is not coded the CADSMCMD first tries to locate the group as a local group and if this fails it looks for a global one.

currentGroup

This option specifies the name of a computer group the software policy is already linked to. If it is coded this link between the current group and the software policy will be removed and replaced by a link between the software policy and the new target computer group.

currentGroupScope

This option specifies the type of group the software policy is already linked to:

global

The computer group has been created at the enterprise manager and maybe replicated to the domain.

local

The computer group has been created at the domain manager.

On enterprise managers all computer groups are global.

If the option is not coded the CADSMCMD first tries to locate the group as a local group and if this fails it looks for a global one.

replace

If this option is specified then the specified group is linked to the specified software policy regardless of another group already being linked to the policy.

seal

If “seal” or “seal=y” is coded then the policy is sealed after the linkage has been completed otherwise it remains unsealed for further modifications. The seal is only performed when the linkage has been successful and no error has been reported. Warnings do not suppress the seal.

Note: If neither the replace option nor the currentGroup option is specified then the action will fail with an error if there is a group already linked to the software policy.

unlink—Unlink Software Policy from Computer Group

This action unlinks a software policy from the computer group it is assigned to. If there is no computer group assigned the command will not report any error or warning and complete successfully. The software policy has to be unsealed otherwise the action will abort with error.

The syntax of the action is as follows:

```
swPolicyaction=unlink
    name=software_policy_name
    [swpScope={distributed | local}]
    [group=computer_group_name
    [groupScope={global | local}]]
```

name

The name of the software policy that is unlinked from the computer group assigned to it.

swpScope

This option determines the type of software policy involved in the action.

«not coded»

If there is a locally created software policy of the required name then this one is taken, if not it is looked for a policy distributed from an enterprise manager. The latter is on domain managers only.

distributed

The specified software policy has to be a policy created at the enterprise manager and distributed to the addressed manager.

local

The specified software policy has to be a locally created policy.

On enterprise manager all software policies have to be of type "local".

group

If this option is coded it is checked whether the computer group to be unlinked from the software policy has the same name as specified with this option. If it matches the action continues otherwise it fails with error. If the option is not coded no such check takes place and the unlink action is processed.

groupScope

If this option is coded then it is checked whether the group associated with the software policy specified is of the group type as specified with this option. If it matches the action is continued otherwise it fails with error. If the option is not coded no such check is performed. The following group types are valid:

global

The computer group has been created at the enterprise manager and maybe replicated to the domain.

local

The computer group has been created at the domain manager.

If the option is not coded the CADSMCMD first tries to locate the group as a local group and if this fails it looks for a global one.

On enterprise managers all computer groups are global.

seal—Seal a Software Policy

This action seals a software policy. If on a domain manager the policy is empty or not assigned to a computer group then the action will fail with error. However, on an enterprise manager it is allowed to seal an unlinked policy that is not empty. If the software policy is already sealed then no error or warning is given and the action terminates successfully.

The syntax of the action is as follows:

```
swPolicyaction=seal
    name=software_policy_name
    [swpScope={distributed | local}]
```

name

The name of the software policy that is sealed

swpScope

This option determines the type of software policy involved in the action.

«not coded»

If there is a locally created software policy of the required name then this one is taken, if not it is looked for a policy distributed from an enterprise manager. The latter is on domain managers only.

distributed

The specified software policy has to be a policy created at the enterprise manager and distributed to the addressed manager.

local

The specified software policy has to be a locally created policy.

On enterprise manager all software policies have to be of type local.

unseal—Unseal a Software Policy

This action unseals a software policy. If the software policy is already unsealed then no error or warning is given and the action terminates successfully.

The syntax of the action is as follows:

```
swPolicyaction=unseal
    name=software_policy_name
    [swpScope={distributed | local}]
```

name

The name of the software policy that is unsealed

swpScope

This option determines the type of software policy involved in the action.

«not coded»

If there is a locally created software policy of the required name then this one is taken, if not it is looked for a policy distributed from an enterprise manager. The latter is on domain managers only.

distributed

The specified software policy has to be a policy created at the enterprise manager and distributed to the addressed manager.

local

The specified software policy has to be a locally created policy.

On enterprise manager all software policies have to be of type "local".

register—Register Software Policy at Domain

This action registers a software policy from the enterprise manager at a domain.

The syntax of the action is as follows:

```
swPolicyaction=register
    {names=software_policy_name}
    [dname=distribution_name]
    [{{area=area_name}|
    {domain=domain_name}| toAllAreas}
    [sendtime=yyyy-mm-dd HH:MM]
    [halttime=yyyy-mm-dd HH:MM]]
    [replace]
    [distPriority={1,...,10}]
```

names

The name of the software policy to be distributed and registered. To process a list of policies the argument can be coded more than once.

dname

Name of the distribution used to send the registration request to the addressed DSM domains. If such a distribution exists and is not yet sealed then the order is added to this distribution. If the distribution is already sealed then the action terminates with error.

If a distribution of the specified name does not exist then it is created.

If this option is not specified a default name is generated and reported by the CLI on stdout.

area

Specifies the name of a DSM domain the registration request is sent to. This parameter can be coded more than once to specify a list of DSM domains to be addressed.

If this option is coded then the associated distribution is sealed and scheduled for processing.

This option should not be coded if the “domain” option or the “toAllAreas” option is coded.

domain

Specifies the name of a DSM domain group the registration request is sent to. This parameter can be coded more than once to specify a list of DSM domain groups to be addressed.

If this option is coded then the associated distribution is sealed and scheduled for processing.

This option should not be coded if the “area” option or the “toAllAreas” option is coded.

toAllAreas

If coded the related distribution is sent to all domain managers registered at the enterprise manager.

If this option is coded then the associated distribution is sealed and scheduled for processing.

This option should not be coded if the “area” option or the “domain” option is coded.

sendtime

Specifies the start time for the processing of the distribution at the enterprise manager. The date is specified in ISO format. If this option is not coded then the actual date/time is taken.

halttime

Specifies the time the processing of distribution is being stopped. The date is specified in ISO format. If this option is not coded then the actual date/time + 1h is taken.

replace

If this option is specified then the software policies are updated. If this option is not coded then the software policies are created, but if they are already exist an error is reported.

distPriority

Specifies the priority assigned to the distribution. The priority is numeric with values from 1 (highest priority) to 10 (lowest priority).
Default: 5

Note: If neither the option “area” nor the option “domain” nor “toAllAreas” is coded then the distribution remains unsealed and further orders can be added to it.

deregister—De-register Software Policy from Domain

This action de-registers a software policy from the enterprise manager at a domain.

The syntax of the action is as follows:

```
swPolicyaction=deregister
    {names=software_policy_name}
    [dname=distribution_name]
    [{{area=area_name}|
    {domain=domain_name}| toAllAreas}
    [sendtime=yyyy-mm-dd HH:MM]
    [halttime=yyyy-mm-dd HH:MM]]
    [distPriority={1,...,10}]
```

names

The name of the software policy to be distributed and de-registered. To process a list of policies the argument can be coded more than once.

dname

Name of the distribution used to send the de-registration request to the addressed DSM domains. If such a distribution exists and is not yet sealed then the order is added to this distribution. If the distribution is already sealed then the action terminates with error.

If a distribution of the specified name does not exist then it is created.

If this option is not specified a default name is generated and reported by the CLI on stdout.

area

Specifies the name of a DSM domain the de-registration request is sent to. This parameter can be coded more than once to specify a list of DSM domains to be addressed.

If this option is coded then the associated distribution is sealed and scheduled for processing.

This option should not be coded if the “domain” option or the “toAllAreas” option is coded.

domain

Specifies the name of a DSM domain group the de-registration request is sent to. This parameter can be coded more than once to specify a list of DSM domain groups to be addressed.

If this option is coded then the associated distribution is sealed and scheduled for processing.

This option should not be coded if the “area” option or the “toAllAreas” option is coded.

toAllAreas

If coded the related distribution is sent to all DSM de-registered at the enterprise manager.

If this option is coded then the associated distribution is sealed and scheduled for processing.

This option should not be coded if the “area” option or the “domain” option is coded.

sendtime

Specifies the start time for the processing of the distribution at the enterprise manager. The date is specified in ISO format. If this option is not coded then the actual date/time is taken.

halttime

Specifies the time the processing of distribution is being stopped. The date is specified in ISO format. If this option is not coded then the actual date/time + 1h is taken.

distPriority

Specifies the priority assigned to the distribution. The priority is numeric with values from 1 (highest priority) to 10 (lowest priority).
Default: 5

Note: If neither the option “area” nor “domain” nor “toAllAreas” is coded then the distribution remains unsealed and further orders can be added to it.

list—List Software Policies

This action lists all software policies registered at the domain or enterprise.

The syntax of the action is as follows:

```
swPolicyaction=list
    [{detailed |separator=separator_sign}]
```

detailed

If this option is coded then a detailed list is provided, otherwise a short list is presented only.

separator

The separator is a character that is used for separating the different output fields in the lines of the listings. The default separator is ":" (colon).

The short list with default separator has the following format:

```
------------------------------------------------------------------
List of Software Policies
Short
------------------------------------------------------------------
«policy_1 name»:«policy_1 type»:«linked group»:«lgroup scope»: [un]sealed
         ...
         ...
         ...
«policy_n name»:«policy_n type»:«linked group»:«group scope»: [un]sealed)
```

The detailed list has the following format:

```
------------------------------------------------------------------
List of Software Policies
Detailed
------------------------------------------------------------------
Policy name     : «name of policy 1»
..Policy type   : {distributed|local}
..State : {disabled|not evaluated|evaluating|not conformant|conformant}
..Priority      : «priority of the template»
..Associated group      : «group name»
..Group type    : {global|local}
..Last evaluated: «yyyy-mm-dd HH:MM»
..Violators     : «number of violators»
..Linkage       : {batch|transaction|
synchronized|no linkage}
..Cascade install       : {yes|no}
```

```
..Degree of automation   : {conformity check|setup jobs|setup and activate jobs}
..Regeneration of jobs   : {yes|no}
..Dynamic group evaluation       : {yes|no}
..Activation time        : «yyyy-mm-dd HH:MM»
..Enterprise time driven : {yes|no}
..Evaluation period      : «evaluation period»
..Evaluation calendar    : «calendar name»
..Adhoc evaluation       : {disabled|on new inventory only|always}
Policy name      : «name of policy 2»
         •
         •
         •
Policy name      : «name of policy n»
```

modify—Modify Attributes of Unsealed Local Software Policy

This command modifies the attributes of an unsealed local software policy. You can optionally seal the policy with this command.

This command has the following format:

```
swPolicyaction=modify
    name=software_policy_name
    [newName=new_name_of_software_template]
    [jcPriority=«1…10»]
    [{transaction|synchronized|nolinkage}]
    [rollback[={y|n}]]
    [cascade[={y|n}]]
    [{nosetup|setup_jobs|setup&activate_jobs}]
    [regenerate[={y|n}]]
    [evaluateGroup[={y|n}]]
    [evaluationStart=YYYY-MM-DD hh:mm]
    [globalTime[={y|n}]]
    [evaluationPeriod=hours]
    [calendarname=calendar_name]
    [{adhoc_disable|adhoc_only|adhoc_always}]
    [comment=new_comment_on_the_software_policy]
    [seal[={y|n}]]
    [userMessage]
```

name

The name of a local software policy that is modified.

newname

Specifies the new name of the selected software policy.

If this option is not coded the policy name is not changed.

jcPriority

This option specifies the new priority for job container generated by this policy. A valid priority is one of the values 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. 1 is the highest priority and 10 is the lowest.

If this option is not coded the priority is not changed.

{transaction|synchronized|nolinkage}

This option specifies the new job linkage for job containers created by the policy.

«not coded»

The job linkage is not changed.

batch

The job linkage is set to batch jobs without transaction.

transaction

The job linkage is set to batch jobs without transaction.

synchronized

The job linkage is set to synchronized job execution.

nolinkage

The job linkage is set to no linkage.

rollback

If the option “rollback” or “rollback=y” is coded then the job linkage is set to "Enable Transaction". If the option is coded as “rollback=n” then “Enable Transaction” is not set.

If this option is not coded the option is not changed.
The parameter should only coded if the job is already in batch processing mode or switched to it by using parameter "transaction" with this command.

Default: rollback=n.

cascade

If the option “cascade” or cascade=y is coded then the SD will resolve the dependencies of cascaded installation request and add the missing installation orders automatically. If the option is coded as “cascade=n” then the dependency is not resolved any longer.

If this option is not coded the option is not changed.

{nosetup|setup_jobs|setup&activate_jobs}

This option drives the results of the policy's evaluation:

«not coded»

The degree of automation is not changed.

nosetup

The degree of automation is changed to checking for conformity only.

setup_jobs

The degree of automation is changed to setting up jobs for non conformant targets but does not activate them.

setup&activate_jobs

The degree of automation is changed to setting up and activating jobs for non conformant targets.

regenerate

If the option "regenerate" or "regenerate=y" is coded then all jobs that failed earlier and have not been completed successfully yet are rescheduled for execution for achieving conformity. If the option "regenerate=n" is coded then this rescheduling is now skipped.

If this option is not coded the option is not changed.

evaluateGroup

If the option "evaluateGroup" or "evaluateGroup=y" is coded then in case of a query group associated with the policy this group will be re-evaluated during the policy's evaluation too. If "evaluationGroup=n" is coded then this re-evaluation of the group will not happen.

If this option is not coded the option is not changed.

evaluationStart

This option specifies a new earliest point in time when the evaluation of the policy will start. The value is specified in ISO format "YYYY-MM-DD hh:mm" where YYYY specifies the year, MM the month, DD the current day of month, hh the hour and mm the minute.

If this option is not coded the option is not changed.

globalTime

If "globalTime" or "globalTime=y" is specified then the time specifications with the policy are based on enterprise time. If "globalTime=n" is coded then the time is domain based.

If this option is not coded the option is not changed.

evaluationPeriod

This option specifies the new period in which the evaluations of the policy takes place. The value is specified in hours. If 0 is specified then the evaluation happens only once at seal.

If this option is not coded the option is not changed.

calendarname

This option specifies the name of a new calendar used by the SD for scheduling the time driven actions of the policy. If an empty string is entered then a possibly calendar entry is removed and calendars are no longer taken into account.

If this option is not coded the option is not changed.

{adhoc_disable|adhoc_only|adhoc_always}

This option specifies the new type of adhoc evaluation:

«not coded»

The adhoc evaluation type does not change.

adhoc_disable

No adhoc evaluation is provided any longer.

adhoc_only

The adhoc evaluation now starts when inventory reports of new or re-installed targets arrive.

ad_hoc_always

The adhoc evaluation now starts at any inventory report from targets.

comment

This option specifies the new comment on the software policy. If an empty string is coded then the old comment is removed.

If the option is not coded the comment remains unchanged.

seal

If “seal” or “seal=y” is coded then the policy is sealed after the modification has been completed otherwise it remains unsealed for further modifications. The seal is only performed when the modification has been successful and no error has been reported. Warnings do not suppress the seal.

If this option is not coded the option is not changed.

userMessage

Modifies the custom administrator message while updating the software policy properties.

showAttr—List Attributes of Software Policy

This action lists the attributes of a software policy.

The syntax of the action is as follows:

```
swPolicyaction=showAttr
    name=software_policy_name
    [swpScope={distributed | local}]
```

name

Name of the software policy of that the attributes will be listed.

swpScope

This option determines the type of software policy involved in the action.

«not coded»

If there is a locally created software policy of the required name then this one is taken, if not it is looked for a policy distributed from an enterprise manager. The latter is on domain managers only.

distributed

The specified software policy has to be a policy created at the enterprise manager and distributed to the addressed manager.

local

The specified software policy has to be a locally created policy.

Note: On enterprise manager all software policies have to be of type "local".

The list of attributes has the following format:

```
---------------------------------------------------------------
Attributes of the Software Policy «software policy name»
---------------------------------------------------------------
Policy type     : {distributed|local}
State   : {disabled|not evaluated|evaluating|not conformant|conformant}
Comment : «comment on the policy»
Priority: «priority of the template»
Associated group: «group name»
Group type      : {global|local}
Last evaluated  : «yyyy-mm-dd HH:MM»
Violators       : «number of violators»
Linkage : {batch|transaction|
synchronized|no linkage}
Cascade install : {yes|no}
Degree of automation    : {conformity check|setup jobs|setup and activate jobs}
Regeneration of jobs    : {yes|no}
Dynamic group evaluation: {yes|no}
Activation time : «yyyy-mm-dd HH:MM»
Enterprise time driven  : {yes|no}
Evaluation period       : «evaluation period»
Evaluation calendar     : «calendar name»
Adhoc evaluation: {disabled|on new inventory only|always}
```

delete—Delete a Software Policy

This action deletes an unsealed software policy.

The syntax of the action is as follows:

```
swPolicyaction=delete
    name=software_policy_name
    [swpScope={distributed | local}]
```

name

Name of the software policy to be deleted.

swpScope

This option determines the type of software policy involved in the action.

«not coded»

If there is a locally created software policy of the required name then this one is taken, if not it is looked for a policy distributed from an enterprise manager. The latter is on domain managers only.

distributed

The specified software policy has to be a policy created at the enterprise manager and distributed to the addressed manager.

local

The specified software policy has to be a locally created policy.

Note: On enterprise manager all software policies have to be of type local.

listViolators—List Violators of Unsealed Software Policy

This action lists all the violators of unsealed software policy. If the policy is not sealed then the request is rejected.

The syntax of the action is as follows:

```
swPolicyaction=listViolators
    name=software_policy_name
    [swpScope={distributed | local}]
    [separator=separator_sign]
```

name

Name of the software policy of that the violators are listed.

swpScope

This option determines the type of software policy involved in the action.

«not coded»

If there is a locally created software policy of the required name then this one is taken, if not it is looked for a policy distributed from an enterprise manager. The latter is on domain managers only.

distributed

The specified software policy has to be a policy created at the enterprise manager and distributed to the addressed manager.

local

The specified software policy has to be a locally created policy.

Note: On enterprise manager all software policies have to be of type local.

separator

The separator is a character that is used for separating the different output fields in the lines of the listings. The default separator is ":" (colon).

The output list with default separator looks as follows:

```
----------------------------------------------------------------
List of Violators for Software Policy «software policy name»
Computer Group Linked: «computer group name»
----------------------------------------------------------------
«name of violator 1»:{Computer | User Profile}
         •
         •
         •
«name of violator n»:{Computer | User Profile}
```

targetcomputer—Target Computer Commands

This method is valid only for domain managers

This section contains the following topics:

General actions

This section contains the following topics:

create—Create Target Computer

This action creates a target computer at the domain manager.

This action has the following format:

```
targetComputer  action=create
    name=computer_name
    computertype={machine|user profile| staging server | docking_device}
    address=address
    [stagingserver=staging_server_name]
    os=os_name
    [calendarname=calendar_name]
    [user= user]
    [phone= phone]
    [location= location]
    [comment=comment] }
    [softwaremanagedsystem [={y/n}] <default = n>]
    [racpolicy={common | disabled | deferred | automatic} <default=common>]
    [hostname=host_name]
    {[macaddress=mac_address] |
    [bootserver=boot_server_name] macaddress=mac_address
    osimage=os_image_name }
```

address

Specifies the network address of the computer.

bootserver

Specifies the name of the boot server the computer is attached to.

calendarname

Specifies the name of the calendar that drives the activities at the target computer

comment

Optional parameter where you can place a comment about the computer.

For instance, information not available in the Computer Attributes or Installation register.

The comment can hold 255 characters.

computertype

An identifier, which can be one of following types:

machine

The specified target is a computer system. It is assumed that a SD agent will be installed on the system.

user profile

Specifies a user profile for a target computer system.
The name of the user profile has the following format:
computer/[domain/]user
For example, a user "john" on system "doe" results in the name "doe/john".
The specified computer must be already defined to DSM when the user profile is created.
The type user_profile replaces the former types "domain_user" and "local_user" which have become obsolete. If they are coded they are mapped to the type user_profile.

Note: The following parameters will be ignored by CADSMCMD when being invoked for ‘targetComputer action=create computerType=“user profile”’:

- Address

- bootServer

- hostName

- macAddress

- os

- osImage

- stagingserver

staging server

The created target computer is a scalability server. It is assumed that SD is installed with this target.

docking device

The created target is a docking device. It is assumed that the address specified is the name of the docking station and that SD is installed with the systems.

hostname

Specifies the host name of the computer.

If not coded, the name of the target computer is used as default.

Note: The parameter “HostName” can only be modified as long as the specified target system has not been registered from the network.
If the system has already been registered, error CMD00149 is reported: “The target has already registered. The operation is not allowed.”

location

If coded, specifies the location of the computer.

macaddress

Specifies the MAC address of the system.

name

Specifies the name of the target.

The name of the target is of one of four types, depending on the value of the "computertype" parameter.

For example, if computertype= is from {machine| staging server| docking device}, then target is a computer system and its name is the name of the computer in the network.

If computertype=user profile, then the name of the user profile is composed of the machine name and the user-id separated by a slash. The user-id can be local machine user-id or a domain user-id.

os

Specifies the operating system type of the target.

For valid values for OS, see Usage of regproc.

The value that is set when creating a target (for instance, Any) will, in current SD server environments, be replaced by the current OS value of the computer when it registers automatically.

Values:

os={AIX|AIX_3.2_RS/6000|AIX_4.1_RS/6000
...|AIX_4.2_RS/6000|AIX_4.3_RS/6000
...|AIX_5.0_RS/6000|AIX_5.1_RS/6000
...|AIX_5.2_RS/6000|AIX_5.3_RS/6000
...|AIX_5.4_RS/6000|AIX_5.5_RS/6000
...|AIX_5.6_RS/6000|AIX_5.7_RS/6000
...|AIX_5.8_RS/6000|ANY|COMPAQ_TRU64_(DEC)
...|COMPAQ_TRU64_5.X_ALPHA
...|DEC_3.0_ALPHA|DEC_3.2_ALPHA
...|DEC_4.0_ALPHA|DGUX|DGUX_4.1_INTEL
...|DGUX_4.1_MC88|DGUX_4.2_INTEL
...|DOS|DRS/NX_7MP_SPARC|DRS/NX_7_SPARC
...|DRS/NX_SPARC|DYNIX|DYNIX_4.2.X_INTEL
...|DYNIX_4.4.X_INTEL|HPUX
...|HPUX_10|HPUX_10_700|HPUX_10_800
...|HPUX_11|HPUX_11I_V2|HPUX_11_700
...|HPUX_11_800|HPUX_11_IA64
...|HPUX_12|HPUX_12I_V2|HPUX_12_700
...|HPUX_12_800|HPUX_12_IA64
...|HPUX_9|HPUX_9_700|HPUX_9_800
...|IRIX|IRIX_6.2|IRIX_6.3
...|IRIX_6.4|IRIX_6.5|LINUX
...|LINUX_2.1_INTEL|LINUX_2.2_INTEL
...|LINUX_2.2_S/390|LINUX_2.3_INTEL
...|LINUX_2.3_S/390|LINUX_2.4_INTEL
...|LINUX_2.4_S/390|LINUX_2.5_INTEL
...|LINUX_2.5_S/390|LINUX_2.6_INTEL
...|LINUX_2.6_INTEL_64-BIT
...|LINUX_2.6_S/390|MACINTOSH
...|MACINTOSH_10.0_PPC|MACINTOSH_10.1_PPC
...|MACINTOSH_10.2_PPC|MACINTOSH_10.3_PPC
...|MACINTOSH_10.4_PPC|NCR_SST
...|NCR_SST_S4I_INTEL|NCR_SST_S4_INTEL
...|NCR_SV|NCR_SV_2X_INTEL
...|NCR_SV_3X_INTEL|NETWARE
...|NETWARE_3_INTEL|NETWARE_4_INTEL
...|NETWARE_5_INTEL|NETWARE_6_INTEL
...|NOKIA_ADMINSUITE|NOKIA_ADMINSUITE_1.0
...|OPENVMS|OPENVMS_5.5_VAX
...|OPENVMS_6.0_ALPHA|OPENVMS_6.0_VAX
...|OPENVMS_6.1_ALPHA|OPENVMS_6.1_VAX
...|OPENVMS_6.2_ALPHA|OPENVMS_6.2_VAX
...|OPENVMS_7.0_ALPHA|OPENVMS_7.0_VAX
...|OPENVMS_7.1_ALPHA|OPENVMS_7.1_VAX
...|OPENVMS_7.2_ALPHA|OPENVMS_7.2_VAX
...|OS/2_16-BIT|OS/2_32-BIT
...|PALM_OS|SCO|SCO_3.2_V5_INTEL
...|SCO_5.2_V5_INTEL|SINIX

```
...|SINIX_5.42_R4000|SINIX_5.43_R4000
...|SINIX_5.44_R4000|SINIX_5.45_R4000
...|SOLARIS|SOLARIS_10_INTEL
...|SOLARIS_10_SPARC|SOLARIS_2.3_SPARC
...|SOLARIS_2.4_INTEL|SOLARIS_2.4_SPARC
...|SOLARIS_2.5_INTEL|SOLARIS_2.5_SPARC
...|SOLARIS_2.6_INTEL|SOLARIS_2.6_SPARC
...|SOLARIS_7_INTEL|SOLARIS_7_SPARC
...|SOLARIS_8_INTEL|SOLARIS_8_SPARC
...|SOLARIS_9_INTEL|SOLARIS_9_SPARC
...|SUNOS|SUNOS_4.1.4_SPARC
...|UNIX|UNIXWARE|UNIXWARE2.0_INTEL
...|UNIXWARE2.1_INTEL|UNIXWARE7.X_INTEL
...|WINDOWS_16-BIT|WINDOWS_32-BIT
...|WINDOWS_95|WINDOWS_98
...|WINDOWS_ME|WIN_2000|WIN_2000_INTEL
...|WIN_CE|WIN_CE_2.00_MIPS
...|WIN_CE_2.00_SH3|WIN_CE_2.01_MIPS
...|WIN_CE_2.01_SH3|WIN_CE_2.11_ARM
...|WIN_CE_2.11_MIPS|WIN_CE_2.11_SH3
...|WIN_CE_2.11_SH4|WIN_CE_3.00_ARM
...|WIN_CE_3.00_MIPS|WIN_CE_3.00_PPC2002_ARM
...|WIN_CE_3.00_SH3|WIN_CE_3.00_SH4
...|WIN_CE_4.20_MOBILE2003_ARM
...|WIN_LONGHORN_SERVER_INTEL
...|WIN_LONGHORN_SERVER_ITANIUM
...|WIN_LONGHORN_SERVER_X64
...|WIN_NT|WIN_NT_ALPHA|WIN_NT_INTEL
...|WIN_NT_MIPS|WIN_SERVER_2003
...|WIN_SERVER_2003_INTEL
...|WIN_SERVER_2003_ITANIUM
...|WIN_SERVER_2003_X64|WIN_VISTA_INTEL
...|WIN_VISTA_ITANIUM|WIN_VISTA_X64
...|WIN_XP|WIN_XP_INTEL|WIN_XP_ITANIUM
...|WIN_XP_X64}
```

Note: For an up-to-date list of supported operating systems and their releases, issue the regproc command or the targetcomputer action=create command in verbose mode.

osimage

Specifies the name of the OS image that is assigned to the system created in a planned configuration.

Before activating the configuration to be installed at the target computer the OS image should become available at the boot server attached to the target computer.

This parameter should only be coded with the macAddress parameter. If the installation order should be deployed in advance also the bootserver should be coded.

phone

If coded, specifies the phone number or email address of the computer user.

racpolicy

Optional parameter which specifies the Reinstall after Crash (RAC) policy of the system.

The following values can be used:

common

The common policy will be used for this computer.

disabled

RAC is disabled on this computer. No job container is set up for this computer. However, its installation records in the installations folder on the SD server will be marked as removed, when the SD agent of the computer reports a new OS installation (by a changed UUID).

deferred

A RAC job container will be created. It can be reviewed and modified, since it has to be activated manually.

automatic

A RAC job container will be created and activated, starting the delivery of job orders to the computer.

Default = common.

Note: If you try to change the RAC policy of a computer when it is in the offline mode, the cadsmcmd targetComputer action=modify command generates an error message.

softwaremanagedsystem

This optional parameter marks a target computer as Software Managed System.

stagingserver

Specifies the name of the scalability server.

If coded the target computer to be created is assumed to be connected through the specified scalability server. The scalability server must already be known to the SD.

Note: If this option is not coded, the Domain Manager is becomes the default scalability server.

user

If coded, specifies the name of the user.

Note: For pre-registration of computers to be managed by OSIM, use the macaddress and osimage with *targetcomputer action=create* command.

Example:

Normally target systems are introduced to the SD servers when they are installing the agent and running the first job check. By this way the new system can only be addressed with software packages when they have been registered from the network.

Sometimes it is better to provide the software for the new systems in advance. Therefore, the new systems have to be added manually to the manager's database.

As soon as they are added they can be taken into account for software templates or job delivery. Nevertheless, the job deliveries will work first when the systems has been registered from the network.

Now assume the system "newComp" should be manually created at the manager and it should meet the following requirements:

The system is attached to the scalability "stage_01".

- The address of the system is newComp.myCompany.com.
- The operating system is of type Windows XP Professional for Intel.
- It is a software managed system.

The following command will introduce this system to the server:

```
cadsmcmd targetcomputer action=create name=newComp address=newComp.myCompany.com
os=WIN_XP_INTEL computerType=machine softwareManagedSystem stagingServer=stage_01
```

delete—Delete Target Computer

This action deletes a target at the domain manager.

This action has the following format:

```
targetcomputer action=delete
    name=computer_name
```

name

Specifies the name of the target to be deleted.

list—List Targetcomputer

This action displays a list of targets known to the system.

An additional filter can be specified to restrict the list to only those targets that fulfill the filter criteria. If no filters are given, the complete list of targets is shown.The filter is limited to 100 bytes.

Note: The filter usage of This action differs from that of other commands.

This action has the following format:

```
targetcomputer action=list
    [{filter=filter|filterfile=file_name}]
    [{DisHwOnly | usrProfOnly | UnMgdPcOnly}]
```

filter

Specifies the expression to confine the amount of information listed

The following attributes are valid to build atomic filter expressions:

Attribute	Type	Compare ops
Computer name	string	=, !=, <, <=, >, >=
Computer type	{ computer | user profile | staging server | docking device}	=, !=
Host name	string	=, !=, <, <=, >, >=
Primary network address	string	=, !=, <, <=, >, >=
IP address	string	=, !=, <, <=, >, >=
Operating system type	any valid os type of the os parameter of the create action	=, !=
Calendar name	string	=, !=, <, <=, >, >=

Attribute	Type	Compare ops
Owner	string	=, !=, <, <=, >, >=
Phone	string	=, !=, <, <=, >, >=
Location	string	=, !=, <, <=, >, >=
Comment	string	=, !=, <, <=, >, >=
staging server name	string	=, !=, <, <=, >, >=
Software managed system	{ yes | no }	=, !=
RAC policy	{ automatic | common | deferred | disabled | offline}	=, !=
Creation date	YYYY-MM-DD hh:mm:ss	=, !=, <, <=, >, >=
Change date	YYYY-MM-DD hh:mm:ss	=, !=, <, <=, >, >=

When building a filter, the attribute names in the atomic expressions are case insensitive.

YYYY-MM-DD hh:mm:ss means a date and time notation in ISO format.

The values for attributes of type string may contain wildcards,

“*” or “?”, where “*” represents an arbitrary even empty string of characters, while “?” represents a single but arbitrary character. If a “*” or “?” should not be interpreted as a wildcard then it should be preceded by an apostrophe (‘). In order to avoid ambiguity, apostrophes as part of the attribute value should be duplicated.

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command..

{DisHwOnly | UsrProfOnly | UnMgdPcOnly}

These options are used to restrict the search on information the user is interested in. If not coded a list of computer, user profiles, scalability servers, docking devices, and OSIM un-managed computers is provided and all related database tables are searched.

DisHwOnly

The list is restricted to registered hardware like computers, scalability servers and docking devices.

UsrProfOnly

The list is restricted to user profiles only

UnMgdPcOnly

The list is restricted to OSIM un-managed computers only.

Example:

To list all targets defined at a domain manager, enter the following command:

```
cadsmcmd targetComputer action=list
```

The provided list may look as follows:

```
--------------------------------------------------------------------------------
List of target computers
--------------------------------------------------------------------------------
PPAPPGP0C (computer; Win NT Intel 4.0.1381)
newComp (computer; Win XP Intel )
HHHHH01B (computer; Win NT Intel 4.0.1381)
KKKKK01A (computer; Win NT Intel 4.0.1381)
stage_01 (staging server; Win 2000 Intel 5.0.2195)
```

```
ORIONXH (computer; Win XP Intel 5.1.2600)
PPPPP98 (computer; Windows 98 4.10.2222)
WWWWW01B (computer; Win 2000 Intel 5.0.2195)
------------------------------------------------------------------------
------------------------------------------------------------------------
List of un-named and BMS un-managed computers
------------------------------------------------------------------------
------------------------------------------------------------------------Total
number of target computers: 8
Number of un-named and BMS un-managed systems shown: 0
SDCMD<A000000>: OK
```

To list all targets defined at a domain manager whose staging server is "stage01-ss" and type "computer" using filters, enter the following command:

```
cadsmcmd targetcomputer action=list filter="(staging server name=stage01-ss &&
Computer type=computer)"
```

The information is listed as follows:

```
------------------------------------------------------------------------
List of target computers
------------------------------------------------------------------------
Test01(computer; WIN_SERVER_2003_INTEL)
------------------------------------------------------------------------
List of un-named and BMS un-managed computers
------------------------------------------------------------------------
------------------------------------------------------------------------
Total number of target computers: 1
Number of un-named and BMS un-managed systems shown: 0
SDCMD<A000000>: OK
```

modify—Modify Target Computers

This action modifies target computer attributes.

This action has the following format:

```
targetcomputer action=modify name=computer_name
    [address=network_address]
    [calendarname=calendar_name]
    [user=user_name]
    [phone=phone]
    [location=location]
    [comment=comment]
    [softwareManagedSystem={y|n}]
    [racpolicy={automatic|common|deferred|disabled}]
    [downloadMethod={NOS|NOSLESS|DTS}]
```

address

Specifies the network address.

This entry should not be the empty string.

calendarname

Specifies the name of the calendar that drives the activities at the target computer

Note: If an empty string is entered no calendar is assumed any longer.

comment

Specifies a comment.

If an empty string is entered, no comment is assumed any longer.

downloadMethod

Optional parameter.

If coded the download method of the addressed target system is switched as follows:

NOS "Internal - NOS"

NOSLESS "Internal - NOS-less"

DTS "DTS - NOS-less"

If not coded the download method remains unchanged.

location

Specifies the location of the computer.

If an empty string is entered, no location is assumed any longer.

name

Specifies the name of the target to be modified.

phone

Specifies the phone number or email address of the user. If an empty string is entered, no phone number or email address is assumed any longer.

racpolicy

Optional parameter which specifies the Reinstall after Crash (RAC) policy of the system.

The following values can be used:

common

The common policy will be used for this computer.

disabled

RAC is disabled on this computer. No job container is set up for this computer. However, its installation records in the installations folder on the SD server will be marked as removed, when the SD agent of the computer reports a new OS installation (by a changed UUID).

deferred

A RAC job container will be created. It can be reviewed and modified, since it has to be activated manually.

automatic

A RAC job container will be created and activated, starting the delivery of job orders to the computer.

Default = common.

Note: If you try to change the RAC policy of a computer when it is in the offline mode, the cadsmcmd targetComputer action=modify command generates an error message.

softwareManagedSystem

If softwareManagedSystem=y, then the system is considered as a Software Managed System. If softwareManagedSystem=n, then it is considered to be not a Software Managed System anymore.

user

Specifies the name of the user.

If an empty string is entered, no user name entry is assumed any longer.

Note: If a parameter is not coded, the related target computer attribute remains unchanged.

Example:

Assume the following attributes of the target computer "newComp" has to be changed:

- The location is "Pdb".
- The user is "joe".
- The users email address is joe@myComp.com

The following command provides the modifications:

```
cadsmcmd targetComputer action=modify name=newComp location=Pdb user=joe
phone=joe@myComp.com
```

showAttr—Show Attributes of Target Computer

This action shows the attributes of the computer system.

This action has the following format:

```
targetcomputer action=showattr
    name=computer_name
```

name

Specifies the name of the target about which the attributes will be shown.

The following information will be shown:

```
--------------------------------------------------------------------------------
Show attributes of target computer
--------------------------------------------------------------------------------
Computer name
Computer type
Host name
Primary network address
IP address
Operating system type
```

```
Owner
Phone
Location
Comment
Staging server name
Calendar name
Software Managed System
RAC Policy
Creation date
Creation time
Change date
Change time
MAC address
Current configuration*
Activated configuration*
  Activation time*
  Status*
  Wakeup*
  Restart*
  Wait for Server*
  Wait for Image*
Planned configuration*
Attached to boot server*
Is boot server
```

** This information is shown if the system is managed by the OS Installation Manager.*

Example:

The following command provides the list of attributes of the target computer "newComp" :

```
cadsmcmd targetComputer action=showattr name=newComp


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "wwwww01b" as user "<default user>" ...OK.
```

```
--------------------------------------------------------------------------------
Show attributes of target computer
--------------------------------------------------------------------------------
Computer name.........................: newComp
Computer type.........................: computer
Network address.......................: newComp.myComp.com
Operating system......................: Win XP Intel
Operating system type.................: WIN_XP
Calendar name.........................:
Owner.................................:
Phone.................................:
Location..............................:
Comment...............................:
Staging server name...................: stage_01
Software Managed System...............: Yes
RAC policy............................: common
Creation date.........................: 2005-04-15
Creation time.........................: 15:43
Change date...........................: 2005-04-15
Change time...........................: 15:43
Is boot server........................: No

sdcmd<a000000>: OK
```

Example for non-ADS devices:

```
--------------------------------------------------------------------------------
Show attributes of target computer
--------------------------------------------------------------------------------
Computer name.........................: mySystem
Computer type.........................: computer
Host name.............................: mySystem
Primary network address...............: mySystem.myCompany.com
IP address............................:
Operating system type.................: ANY
Owner.................................:
Location..............................:
Phone.................................:
Comment...............................:
Staging server name...................: my-DSM-stage
Calendar name.........................:
Software Managed System...............: No
RAC policy............................: common
Creation date.........................: 2006-09-20
Creation time.........................: 16:42:39
Change date...........................: 2006-09-20
Change time...........................: 16:42:41
MAC address...........................: FF:01:23:45:67:8A
Is ADS device.........................: No
Current configuration.................:
Activated configuration...............:
Planned configuration.................: myWinXP
Attached to boot server...............: my-DSM-stage
Is boot server........................: No
```

Example showing the attributes of an ADS managed system:

```
--------------------------------------------------------------------------------
Show attributes of target computer
--------------------------------------------------------------------------------
Computer name.........................: name=myADSdevice
Computer type.........................: computer
Host name.............................: name=myADSdevice
Primary network address...............: name=myADSdevice.myCompany.com
IP address............................:
Operating system type.................: ANY
Owner.................................:
Location..............................:
Phone.................................:
Comment...............................:
Staging server name...................: myManager
Calendar name.........................:
Software Managed System...............: No
RAC policy............................: common
Creation date.........................: 2006-09-20
Creation time.........................: 13:57:12
Change date...........................: 2006-09-20
Change time...........................: 13:57:14
MAC address...........................: FF:01:23:45:67:89
Is ADS device.........................: Yes
ADS server name.......................: myADSserver.ADSnet.myCompany.com
Is boot server........................: No
```

Example showing the attributes of an OSIM managed system:

```
012: targetcomputer action=showattr name=HUMPFA
Command is processed ...
------------------------------------------------------------------------------
Show attributes of target computer
------------------------------------------------------------------------------
Computer name.........................: HUMPFA
Computer type.........................: computer
Network address.......................: HUMPFA.ca.com
Operating system......................: Windows
Operating system type.................: WINDOWS_16-BIT
Calendar name.........................:
Owner.................................:
Phone.................................:
Location..............................:
Comment...............................:
scalability server name...............:
Software Managed System...............: No
RAC policy............................: common
Creation date.........................: 2005-03-27
Creation time.........................: 11:04
Change date...........................: 2005-03-27
Change time...........................: 11:04
MAC address...........................: 11.22.33.44.55.77
Current configuration.................:
Activated configuration...............: w2kp_ger
..Activation time.....................: 2005-03-27 11:00:47
..Status..............................: stopped
..Wake up.............................: 1
..Restart.............................: 1
..Wait for server.....................: 0
..Wait for image......................: 0
Planned configuration.................:
Attached to boot server...............:
Is boot server........................: No

sdcmd<A000000>: OK
```

The wake up attribute indicates, if a wake up is to be performed or not, and the restart attributes indicates, if a restart is automatically scheduled or not.

If wakeup or restart are specified, then "1" is shown, otherwise "0".

If a “Wait for server” shows “Required” then the OS installation request is deferred until the target is assigned to a boot server.

If “Waiting” is shown then the request has fallen due but is deferred because of the missing boot server.

If the entry “Wait for server” is missing then either no delay has been required or a boot server is already assigned.

If a “Wait for image” shows “Required” then the OS request is deferred until the related boot and OS images are staged at the related boot server.

If “Waiting for boot image” is shown then the related boot image is not staged at the boot server and the request is deferred although it has fallen due already.

If the “Waiting for OS image” is shown then the related OS image is not yet staged at the bootserver and the request is deferred although it has fallen due already.

If the entry “Wait for image” is not shown then either no request for delay has been specified with the request or all required images are available at the boot server.

Job Management Commands

This section contains the following topics:

activateJobCheck—Activate the Job Check Program

This actions start the Job Check program making the agent in the selected computer contact its domain manager to check if there are any jobs waiting for it and to report changes in computer attributes.

It will not wait for its completion.

The action has the following format:

```
targetcomputer action=activateJobCheck
    computer=name
```

computer

Specifies the name of the computer where the Job Check program will be started.

activateStageCheck—Activate the Stage Check Program

This actions starts the Stage Check program that initiates a check for pending orders.

It will not wait for its completion.

This action has the following format:

```
targetcomputer action=activatestagecheck
    computer=name
```

computer

Specifies the name of the computer where the Stage Check program will be activated.

listFailedJobs—List Failed Jobs Related to Target Computer

This action lists all failed jobs related with this target computer. To confine the amount of data, a filter can be specified.

This action has the following format:

```
targetcomputer action=listFailedJobs
    name=computer_name
    [{alt |
      {filter=filter|filterfile=file_name}
    [detailed]}]
```

alt

An alternative output list is provided.

An alternate list is provided based on the item name and version only.

In case of coding this parameter the value of the item and version parameter might contain database supported wild cards, for example,

"%"

for an arbitrary, even empty sequence of characters

"_"

for an arbitrary single character

For Microsoft SQL Server also the square brackets ("[","]") have a special meaning. This has to be taken into account when coding the name.

detailed

Information is listed in detail

filter

Specifies the expression to confine the amount of information listed

For attributes valid to build atomic filter expressions see action "listjobs".

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command.

name

Specifies the name of the computer of which the failed jobs will be listed.

Examples

Launching the following command

```
cadsmcmd targetComputer action=listFailedJobs name=comp1
```

results in the following output:

```
--------------------------------------------------------------------------------
List of failed jobs for computer "comp1"
--------------------------------------------------------------------------------

tstkpk02 1.0:instCrash (EXECUTION_ERROR)

Total number of jobs read: 5
Total number of jobs shown 1
SDCMD<A000000>: OK
```

When launching the command

```
cadsmcmd targetComputer action=listFailedJobs name=comp1 detailed
```

then the following list is received:

```
-------------------------------------------------------------------------------
List of failed jobs for computer "comp1"
(detailed)
-------------------------------------------------------------------------------
Computer job identifier.............: 0C42D0D1-3E32-4974-8F47-E7220805BC4A
..Job name..........................: tstkpk02 1.0:instCrash
..Item name.........................: tstkpk02
..Item version......................: 1.0
..Procedure name....................: instCrash
..Computer installation procedure...: instCrash
..Ordered by........................: winnt://comp1/administrator
..Created at........................: 2006-10-19 15:41
..Started at........................: 2006-10-19 15:41
..Terminated at.....................: 2006-10-19 15:41
..State.............................: EXECUTION_ERROR
..Error cause.......................: 228001
..Error message.....................: Exit code 2 indicates possible error

..Job output........................: No output available for this job.

Total number of jobs read: 5
Total number of jobs shown 1
SDCMD<A000000>: OK
```

And when launching the following command

```
cadsmcmd targetComputer action=listFailedJobs name=comp% alt
```

then the following list is received:

```
-------------------------------------------------------------------------------
List of failed jobs "comp%"
-------------------------------------------------------------------------------
Target..............................: comp0
..Job name..........................: tstkpk01 1.0: inst
..Job identifier....................: 23131B7D-AAE2-4878-A33B-563E473FF37E
..Item name.........................: tstkpk01
..Item version......................: 1.0
..Procedure name....................: inst
..Task..............................: install
..User parameters...................:
```

```
..Job state.........................: DELIVERY_ERROR
..Error code........................: 228432
..Error message.....................: Sending message to computer <130.11
....................................+ 9.101.163> failed.
..Creation date.....................: 2006-10-20
..Creation time.....................: 17:16
..Activation date...................: 2006-10-20
..Activation time...................: 17:15
..Completion date...................: 2006-10-20
..Completion time...................: 17:16
Target..............................: comp1
..Job name..........................: tstkpk02 1.0:instCrash
..Job identifier....................: 0C42D0D1-3E32-4974-8F47-E7220805BC4A
..Item name.........................: tstkpk02
..Item version......................: 1.0
..Procedure name....................: instCrash
..Task..............................: install
..User parameters...................:
..Job state.........................: EXECUTION_ERROR
..Error code........................: 228001
..Error message.....................: 2
..Creation date.....................: 2006-10-19
..Creation time.....................: 15:41
..Activation date...................: 2006-10-19
..Activation time...................: 15:41
..Completion date...................: 2006-10-19
..Completion time...................: 15:41


--------------------------------------------------------------------------------
Total number of items: 2
SDCMD<A000000>: OK
```

To receive the list of failed jobs for computer "test01" with job state as "Execution_Error" using filters, enter the following command:

```
cadsmcmd targetcomputer action=listfailedjobs name=test01 filter="Job
state=EXECUTION_ERROR"
```

The output is as follows:

```
------------------------------------------------------------------------
List of failed jobs for computer "test01"
------------------------------------------------------------------------
CA DSM Agent + AM, RC, SD plugin(s) (ENU) Win32 12.0.0.4860:Install (EXECUTION_ERROR)
CA DSM Agent + AM, RC, SD plugin(s) (ENU) Win32 12.0.0.4860:Install (EXECUTION_ERROR)
CA DSM Agent + AM, RC, SD plugin(s) (ENU) Win32 12.0.0.4860:Install (EXECUTION_ERROR)

Total number of jobs read: 16
Total number of jobs shown 3
SDCMD<A000000>: OK
```

listInst—List Installed Products

This action lists the installations at the target computer.

This action has the following format:

```
targetcomputer action=listinst
    name=computer_name
    [detailed]
```

detailed

Information is listed in detail

name

Specifies the name of the computer of which the software installations will be listed.

Example:

Enter the following command to provide a short list of all installations at the target computer "KKKKK01B".

```
cadsmcmd targetComputer action=listInst name=KKKKK01B


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
List of installations on target computer "KKKKK01B"
--------------------------------------------------------------------------------

SD Scalability Server for WinNT/W2K 4.0 ENU (I386): Scalability Server (EXECUTION_OK)
CA ITCM Explorer for WinNT/W2K 4.0 ENU (I386): Local (EXECUTION_OK)
SD Agent for WinNT/W2K 4.0 ENU (I386): Agent (EXECUTION_OK)
SD Server for WinNT/W2K 4.0 ENU (I386):  (EXECUTION_OK)
tstbase 1.0: inst (EXECUTION_OK)
Data Transport Service for Win32 2.0 SP1 ENU: DTS (EXECUTION_OK)
tstkpk03 1.0: inst (EXECUTION_OK)
tstkpk02 1.0: instCrash (EXECUTION_OK)
--------------------------------------------------------------------------------
Total number of installations found: 8
Number of installations shown: 8
SDCMD<A000000>: OK
```

To retrieve a more detailed list enter the following command:

```
cadsmcmd targetComputer action=listInst name=KKKKK01B detailed


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "wwwww01b" as user "<default user>" ...OK.

-------------------------------------------------------------------------------
List of installations on target computer "KKKKK01B"
(detailed)
-------------------------------------------------------------------------------
Computer job identifier.............: 5107
..Job name..........................:
..Item name.........................: SD scalability server for WinNT/W2K
..Item version......................: 4.0 ENU (I386)
..Procedure name....................: scalability server
..Ordered by........................: System
..Created at........................: 2005-04-15 10:42
..Started at........................: 2005-04-15 10:42
..Installed at......................: 2005-04-15 10:42

Computer job identifier.............: 5113
..Job name..........................:
..Item name.........................: CA ITCM Explorer for WinNT/W2K
..Item version......................: 4.0 ENU (I386)
..Procedure name....................: Local
..Ordered by........................: System
..Created at........................: 2005-04-15 10:42
..Started at........................: 2005-04-15 10:42
..Installed at......................: 2005-04-15 10:42
```

```
Computer job identifier.............: 5110
..Job name..........................:
..Item name.........................: SD Agent for WinNT/W2K
..Item version......................: 4.0 ENU (I386)
..Procedure name....................: Agent
..Ordered by........................: System
..Created at........................: 2005-04-15 10:42
..Started at........................: 2005-04-15 10:42
..Installed at......................: 2005-04-15 10:42

Computer job identifier.............: 5104
..Job name..........................:
..Item name.........................: SD Server for WinNT/W2K
..Item version......................: 4.0 ENU (I386)

Computer job identifier.............: 5418
..Job name..........................: tstbase 1.0: inst
..Item name.........................: tstbase
..Item version......................: 1.0
..Procedure name....................: inst
..Ordered by........................: KKKKK01
..Created at........................: 2005-04-15 10:59
..Started at........................: 2005-04-15 10:57
..Installed at......................: 2005-04-15 10:59

Computer job identifier.............: 5510
..Job name..........................:
..Item name.........................: Data Transport Service for Win32
..Item version......................: 2.0 SP1 ENU
..Procedure name....................: DTS
..Ordered by........................: Software detector
..Created at........................: 2005-04-15 10:59
..Started at........................: 2005-04-15 10:59
..Installed at......................: 2005-04-15 10:59

Computer job identifier.............: 6159
..Job name..........................: tstkpk03 1.0: inst
..Item name.........................: tstkpk03
..Item version......................: 1.0
..Procedure name....................: inst
..Ordered by........................: KKKKK01
..Created at........................: 2005-04-15 11:23
..Started at........................: 2005-04-15 11:22
..Installed at......................: 2005-04-15 11:23
```

```
Computer job identifier.............: 7805
..Job name..........................: tstkpk02 1.0:instCrash
..Item name.........................: tstkpk02
..Item version......................: 1.0
..Procedure name....................: instCrash
..Ordered by........................: KKKKK01
..Created at........................: 2005-04-15 12:15
..Started at........................: 2005-04-15 12:15
..Installed at......................: 2005-04-15 12:15


--------------------------------------------------------------------------------
Total number of installations found: 8
Number of installations shown: 8
SDCMD<A000000>: OK
```

listJobs—List Jobs Related to Target Computer

This action lists those jobs, which are related to the system identified in the name parameter. An additional filter can be specified to restrict the list to only those jobs that fulfill the filter criteria. If no filters are given the complete list of jobs is shown. The filter is limited to100 bytes.

This action has the following format:

```
targetcomputer action=listjobs
    name = computer_name
    [{alt |
    [{filter=filter|filterfile=file_name}]]
```

alt

An alternative output list is provided.

An alternate list is provided based on the item name and version only.

In case of coding this parameter the value of the item and version parameter might contain database supported wild cards, for example,

"%"

for an arbitrary, even empty sequence of characters

"_"

for an arbitrary single character

For Microsoft SQL Server also the square brackets ("[","]") have a special meaning. This has to be taken into account when coding the name.

filter

Specifies the expression to confine the amount of information listed

The following attributes are valid to build atomic filter expressions:

Attribute	Type
Job name	String
Job state	{UNDEFINED | WAITING | DELIVERY_ORDERED | DELIVERING | DELIVERY_OK | DELIVERY_ERROR | ALREADY_DELIVERED | EXECUTION_ORDERED | EXECUTING | EXECUTION_OK | EXECUTION_ERROR | UNDELIVERY_ORDERED | UNDELIVERING | UNDELIVERY_OK | UNDELIVERY_ERROR | ALREADY_INSTALLED | MANIPULATION_NOT_ALLOWED}
Item name	String
Item version	String
Item type	{SOFTWARE | DOCUMENT | PROCEDURE | UNSEALED ITEM | ENABLED ITEM | DETECTED ITEM | ARCHIVED ITEM}
Procedure name	String
Ordered by	String
Error message	String
Error cause	Long
Comment	String
Computer installation procedure	String
Creation date	Sdate
Creation time	Stime
Activation date	Sdate
Activation time	Stime
Completion date	Sdate
Completion time	Stime

filterfile

Specifies the name of the file that contains the filter commands.

The commands in the file are specified in the same way as in the filter parameter, but the size of the file is limited by the capacity of the system that runs this command.

name

Specifies the name of the computer of which the jobs will be shown.

Example:

When launching the following command

```
cadsmcmd targetComputer action=listJobs name=sys1
```

then the following list might return:

```
--------------------------------------------------------------------------------
List of jobs for computer sys1
1. List of current installations
--------------------------------------------------------------------------------
CA Unicenter CA ITCM Explorer 11.0.8070.296:Install (EXECUTION_OK,
B8E5FB21-0681-4C97-8FC1-E7B2E718B9CA)
CA Unicenter DSM Agent + Software Delivery Plugin 11.0.8070.296:Install
(EXECUTION_OK, C046CFB6-F2F9-4D05-B7E9-006E47F42454)
CA Unicenter DSM Agent + Asset Management Plugin 11.0.8070.296:Install (EXECUTION_OK,
EBC708F8-39C9-405A-B69D-F65DA74A72F6)
CA Unicenter DSM Agent + Basic Inventory Plugin 11.0.8070.296:Install (EXECUTION_OK,
97741439-351E-4D03-9C64-33775C99649C)
tstbase 1.0:inst (EXECUTION_OK, FC23989D-C88B-4ADB-8A2E-30223A65B71E)
```

```
--------------------------------------------------------------------------------
2. List of other jobs
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Total number of jobs: 5
Total number of jobs read: 5
Total number of jobs shown 5
SDCMD<A000000>: OK
showing job status and job id in parentheses.
And when launching the following command
cadsmcmd targetComputer action=listJobs name=sys% alt
then the following list might return:
--------------------------------------------------------------------------------
List of jobs "sys%"
--------------------------------------------------------------------------------
Target.................................: sys1
..Job name.............................: tstbase 1.0: inst
..Job identifier.......................: FC23989D-C88B-4ADB-8A2E-30223A65B71E
..Item name............................: tstbase
..Item version.........................: 1.0
..Procedure name.......................: inst
..Task.................................: install
..User parameters......................:
..Job state............................: EXECUTION_OK
..Error code...........................: 0
..Error message........................:
..Creation date........................: 2006-10-18
..Creation time........................: 13:58
..Activation date......................: 2006-10-18
..Activation time......................: 13:57
..Completion date......................: 2006-10-18
..Completion time......................: 13:59

Target.................................: sys2
..Job name.............................: tstbase 1.0: inst
..Job identifier.......................: 653C4ECE-10BD-43AF-9809-3DAC494A35D8
..Item name............................: tstbase
..Item version.........................: 1.0
..Procedure name.......................: inst
..Task.................................: install
```

```
..User parameters.....................:
..Job state...........................: EXECUTING
..Error code..........................: 228432
..Error message.......................: File transfer has completed successful
.....................................+ ly
..Creation date.......................: 2006-10-19
..Creation time.......................: 09:08
..Activation date.....................: 2006-10-19
..Activation time.....................: 09:08
..Completion date.....................: 2006-10-19
..Completion time.....................: 09:08


------------------------------------------------------------------------
Total number of items: 2
SDCMD<A000000>: OK
```

When launching the command using filters,

```
cadsmcmd targetcomputer action=listjobs name=test01 filter="Job state=EXECUTION_OK"
```

The output is as follows:

```
------------------------------------------------------------------------
List of jobs for computer test01
1. List of current installations
------------------------------------------------------------------------
CA DSM Scalability Server 12.0.0.4860:Install (EXECUTION_OK, 21EC3E23-ABA8-4C18-
8683-ABA423928815)
CA DSM Explorer 12.0.0.4860:Install (EXECUTION_OK, F4B97228-9BC0-458D-9378-6EF46
FDAFBA6)
CA DSM Agent + Asset Management plugin (English only Edition) 12.0.0.4860:Instal
l (EXECUTION_OK, 3643F431-59C8-4AEC-B213-66813863F053)
CA DSM Agent + Basic Inventory plugin (English only Edition) 12.0.0.4860:Install
 (EXECUTION_OK, 224D083A-D48F-4EAE-8060-37B6486580F5)
CA DSM Agent + Software Delivery plugin (English only Edition) 12.0.0.4860:Insta
ll (EXECUTION_OK, 742B5764-E4AF-475A-BA99-08B380BF6DC0)
CA DSM Agent + Data Transport plugin 12.0.0.4860:Install (EXECUTION_OK, 11A8AF71
-F703-44BF-B3BA-40D1C3218560)
```

```
CA DSM Agent + Remote Control plugin (English only Edition) 12.0.0.4860:Centrall
y Managed Complete Agent (EXECUTION_OK, 94A845B7-ACEE-4AF2-8760-6C4F0EFDA570)
CA DSM Manager 12.0.0.4860:Detect (EXECUTION_OK, 2A8EDFFE-7B1E-4229-94B1-4AD816E
9FF11)

------------------------------------------------------------------------------
2. List of other jobs
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Total number of jobs: 11
Total number of jobs read: 11
Total number of jobs shown 8
SDCMD<A000000>: OK
```

listMembership—List Memberships of TargetComputer

This action lists all the groups that the specified target computer is a member of.

This action has the following format:

`targetcomputer action=listMembership name=`*`computer_name`*

name

Specifies the name of the target computer. The memberships will be listed.

Example:

Enter the following command to list all those groups the target computer "KKKKK01B" is a member of:

```
cadsmcmd targetComputer action=listMembership name=KKKKK01B
```

This might result in the following list:

```
------------------------------------------------------------------------------
"KKKKK01B" is member of the following Groups
------------------------------------------------------------------------------
cg (Group, local)
qg (Query group, global)
Software Delivery Server (Group, local)
tqg (Query group , global)
SDCMD<A000000>: OK
```

reinstallProduct—ReInstall Product

This action reinstalls a product on a target computer that is already marked as installed.

This action has the following format:

```
targetcomputer action=reInstallProduct
    name=computer_name
    item=item_name
    version=item_version
    procedure=install_procedure
    cname=[name]
    [{noLinkage|rollback|synchronized}]
    [nocascade]
    [deliveryTime="YYYY-MM-DD hh:mm"]
    [atTime="YYYY-MM-DD hh:mm"]
    after={exacttime|boottime}
    [calendarname=delivery_calendar_name]
    [preaction={none|reboot|logoff}]
    [postaction={none|reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}]
    [promptUser[={y|n}]]
    [allowCancel[={y|n}]]
    [execTimedOut[={y|n}]]
    [prompt=days.hours]
    [offline[={y|n}]]
    [runAtShutdown[={y|n}]]
    [preventLogon[={y|n}]]
    [globalTime]
    [noCalendar]
    [parameters=user_parameters]
    [jobTimeOut=d.h]
    [unsealed]
```

after

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

allowCancel

If "allowCancel" or "allowCancel=y" is coded, the user will be granted to cancel the job's execution.
If "allowCancel=n" is coded, the user will not be granted.

If the parameter is not coded, the default is given by the related procedure's job option value.

If neither promptUser is set by this call nor already enabled, then this parameter is ignored. No warning is given.

atTime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

calendarName

Specifies the name of the calendar that controls the time when the evaluation can take place.

cname

Specifies an the optional unique name of a job container (default: *target_computer_name*[*YYYY-MM-DD hh:mm:ss*]).

It is used for distributing the request. If the container already exists, but is unsealed, it is used without changing its linkage or cascade options. If it exists and is sealed, an error is reported. If it does not exist, the job container will be created according to the linkage and cascade parameters specified.

deliveryTime

Specifies the start time of delivery from the domain manager.
The date has the ISO format "YYYY-MM-DD hh:mm".

If the parameter is not coded, the actual date and time are used.

execTimedOut

If "execTimedOut " or "execTimedOut=y" is coded, the job's execution will be automatically started when the user prompt times out.
If "execTimedOut=n", the job will not automatically be started.

If neither promptUser is set by this call nor already enabled, then this parameter is ignored. No warning is given.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

item

Specifies the name of the item to be reinstalled.

jobTimeout

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

name

Specifies the name of the computer where the product should be reinstalled.

noCalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

noCascade

The job linkage option is set to "Ignore cascading".

This parameter is allowed only if "transaction" or "synchronized" are coded.

noLinkage|rollback|synchronized

In case of more than one job, this mode handles the linkage between the different jobs of a job container. The coding has the following effects:

none coded

Batch job mode used without rollback.

noLinkage

Run the job independently of the other jobs in the container.

rollBack

Set the job linkage option to "Enable Transaction".

synchronized

The job linkage option is set to "Synchronized job execution".

offline

If "offline" or "offline=y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.
If "offline=n" the connection will not be released.

If the parameter is not coded, the default is given by the related procedure's job option value.

parameters

Specifies the user parameters for the procedure.
Multiple user parameters must be separated by spaces inside double quotes.
If an empty string is coded, then no parameters are assumed.

postaction

Specifies the necessary actions that must take place after completion of the specified procedure. The following values are valid:

none

Performs no post-action.

reboot

Restarts the system after completion of the procedure.

logoff

Logs off the user after completion of the procedure.

rebootAtEnd

Restarts the system after completion of all the jobs of this container.

logoffAtEnd

Logs off the user after completion of all the jobs of this container.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

If the parameter is not coded, the default is given by the related procedure's job option value.

preaction

Specifies that necessary actions must take place before the specified procedure starts. The following values are valid:

none

Performs no pre-action.

reboot

Restarts the system before the start of the procedure.

logoff

Logs off the user before the start of the procedure.

preventLogon

If "preventLogon" or "preventLogon=y" is coded, then the user logon will be rejected during the job's runtime. If a user is already logged on, the execution of the job is delayed until the user logs off.
If "preventLogon=n" the user logon will not be rejected.

procedure

Specifies the name of a procedure

procedure to be used for the reinstallation.

prompt=d.h

Specifies the time period in which the user will be prompted for the job's start.

The period's format is "*d.h*" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0). If the specified value falls below the valid minimum, the value is replaced by the minimum. If the specified value exceeds the valid maximum, then the value is replaced by the maximum. No warning is given.

If neither promptUser is set by this call nor already enabled, then this parameter is ignored. No warning is given.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

If the parameter is not coded, the default is given by the related procedure's job option value.

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

unsealed

The addressed job container remains unsealed after the generated job has successfully been created.

version

Specifies the version of the item.

The parameters "allowCancel," "execTimedOut" and "prompt" are valid, if the "promptUser" option is already set or is set by this command using the "promptUser" parameter.
If neither the "promptUser" option of a job nor the "promptUser" parameter of the command is set, these parameters will be ignored.
No warning is given.

removeJob—Remove Job from Job List

This action removes jobs from the specified target computer's job list.

This action has the following format:

```
targetcomputer action=removeJob name=computer_name
    {cjid=computer_job_identifier|
    item=item_name
    version=item_version
    procedure=item_procedure
    [installedWith=installation_procedure]}
```

cjid

Specifies the identifier of the computer job to be removed.

installedWith

Specifies the name of the install procedure referred to in the job to be removed.

item

Specifies the name of the item that the job to be removed deals with.

name

Specifies the name of the computer for which the job is removed from its job list.

procedure

Specifies the name of the item procedure referred to in the job to be removed.

version

Specifies the version of the item the job to be removed deals with.

Note: If the job to be removed is specified by item, version, and so on, the first job found in the job list that matches the criteria will be removed only.

Example:

To remove the job "tstkpk01 1.0:inst" of job id "2F126122-3848-4FE7-A41F-DF781E7CB5AD6050" from the list of associated jobs of the target computer "KKKKK01B" enter the following command:

```
cadsmcmd targetComputer action=removeJob name=KKKKK01B
cjid=2F126122-3848-4FE7-A41F-DF781E7CB5AD6050
```

showJobAttr—Show Job Attributes

This action shows the attributes of a job related to a computer. The related computer is indicated by the name parameter and the job in question is indicated by the jobname parameter.

This action has the following format:

```
targetcomputer action=showjobattr
    name=computer_name
    jobname=job_name
```

jobname

Specifies the name of the job for which the attributes will be shown.

name

Specifies the name of the computer related to the job attributes to be shown.

Example:

To show the attributes of the job “tstkpk02 1.0:inst” of KKKKKB1B just invoke:

```
cadsmcmd targetComputer action=showJobAttr name=KKKKK01B jobName=”tstkpk02
1.0:inst”
```

and the information presented might look as follows:

```
--------------------------------------------------------------------------------
Show attributes of job "tstkpk02 1.0:inst"
.........of computer "KKKKKB1B"
--------------------------------------------------------------------------------
Computer job identifier.............: AF60EB2F-7706-4B78-B084-6B09DCB96D94
..Job name..........................: tstkpk02 1.0: inst
..Item name.........................: tstkpk02
..Item version......................: 1.0
..Item type.........................: SOFTWARE
..Task.............................: 1
```

```
.....................................install
..Procedure name.....................: inst
..Ordered by.........................: winnt://677-lab-xxx2/xxx
..Job state..........................: EXECUTION_OK
..Error message......................: OK
..Error cause........................: 0
..Comment............................:
..Computer installation procedure....: inst
..Activation date....................: 2006-08-16
..Activation time....................: 15:01
..Creation date......................: 2006-08-16
..Creation time......................: 15:02
..Completion date....................: 2006-08-16
..Completion time....................: 15:03
..Data sent in per cent..............:
..Job output.........................: No output available for this job.

SDCMD<A000000>: OK
```

TargetComputer Commands Related to OS Installation Management (OSIM)

OS Installation Management (OSIM) is the feature of Software Delivery which deals with the installation of boot images and operating system images from scratch and with their reinstallation after a system crash. The methods listed following describe how the management functionality of OSIM is provided by cadsmcmd.

OSIM administers all data about operating system installations in a network or sub-network.

Note: For

- Create Target Computer
- Delete Target Computer
- List Target Computers
- Show Attributes

see section General Commands (see page 384)

This section contains the following topics:

activateOS—Activate OS Installation Order

This action schedules a planned installation of an operating system for execution.

If the target computer is OSIM managed, a planned operating system is available for it, and if there is no OS installation pending for it, then the installation of the planned OS is scheduled for execution.

In any other case, the action will fail.

This action has the following format:

```
targetComputer action=activateOS
    name=computer_name
    [atTime="YYYY-MM-DD hh:mm"]
    [wakeup[={y|n=}]]
    [restart[={y|n}]]
    [waitBs[={y|n}]]
    [waitIm[={y|n}]]
```

attime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

name

Specifies the name of the computer for which the installation of the planned operating system is scheduled for execution.

restart

This optional parameter specifies if a running system will be restarted when the associated installation request will be carried out.

If restart or restart=y is coded, then a restart will be performed.

If restart=n is coded, then no automatic restart will be performed and it is up to the user to reboot the system to start the OS installation process.

If the parameter is not coded, the default is given by restart=n.

waitBs

If waitBs or waitBs=y is coded then the activation request might be deferred until the addressed target is assigned a boot server. If it waitBs=n is coded then the request is not deferred because of a missing boot server assignment. The default is waitBs=n

WaitIm

If waitIm or waitIm=y is coded then the activation request might be deferred until the associated boot and OS images are staged at the related boot server. If it waitIm=n is coded then the request is not deferred because of not yet staged images. The default is waitIm=n.

wakeup

This optional parameter specifies whether a target system in power standby mode shall be waked up for this installation or not.
If wakeup or wakeup=y is coded, then the systems will be waked up.
If wakeup=n is specified, no wake up will be performed.
If the parameter is not coded, the default wakeup=n will be used.

Example:

Assume that a planned installation of an operating system should be processed for target computer dorabella. If the system is in power standby mode it should be waked up. Furthermore an automatic restart should be performed to start the installation process at the target system. The following command will provide this request:

```
cadsmcmd targetcomputer action=activateOS name=dorabella wakeup restart
```

cancelOS—Cancel OS Installation Order

This action cancels a pending installation of an OS image.

If the specified target system exists, and if it is OSIM managed, and if an OS installation order is pending for it, this method will cancel this installation request. In any other case, the method will fail.

Note: A pending installation can only be canceled as long as the order is scheduled at the boot server and the target system has not started with the installation yet.

This action has the following format:

```
targetComputer action=cancelOS
    name=computer_name
```

name

Specifies the name of the computer of which the pending installation will be canceled.

deletePlannedOS—Delete Planned OS Installation Order

This action allows you to delete a planned configuration.

If the system is OSIM managed and if there is a planned OS installation order, then this planned OS installation is deleted. In any other case, the request will fail.

If the planned configuration to be removed is the only configuration associated with the specified system the system will become OSIM unmanaged.

This action has the following format:

```
targetComputer action=deletePlannedOS
    name=computer_name
```

name

Specifies the name of the computer of which an existing planned configuration will be deleted.

deleteScheduledOS—Delete Scheduled OS Installation Order

This action deletes a scheduled OS installation request of sub-state "failed" or "stopped".

If the system is OSIM managed and if there is a scheduled OS installation order of required sub-state, then this scheduled OS installation order is deleted. In any other case, the request will fail.

If the scheduled configuration to be removed is the only configuration associated with the specified system the system will become OSIM unmanaged.

This action has the following format:

```
targetComputer action=deleteScheduledOS
    name=computer_name
```

name

Specifies the name of the computer of which an existing scheduled configuration will be deleted.

modifyInstallParameter—Modify Install Parameters of OS Installation Orders

This action modifies installation parameters of a OSIM managed system for a planned OS installation order.

The specified parameter is modified according to the arguments specified. If default is coded the parameter is reset to the default value, if there is any. If there is none an error is reported.

Otherwise, the old parameter value is replaced by the specified one if the new value fits in the parameter specification of the related OS image.

It is not possible to change the OS image parameters by this action. A different action has to be chosen for changing the OS image assignment (action modifyOS).

When the MAC address is modified, not only the configuration parameter is changed but also the computer property. It is checked if another computer in the OSIM database already uses the MAC address. If there is any named computer entry of this MAC address, that is not queued for deletion yet, the changing of the MAC address will be rejected. If there is an unnamed computer entry of the MAC address this one will be deleted from database.

Modifying a parameter of type "extendable list" by a parameter value that is not a member of the associated list will add this value to the list. For non-extensible lists the value has to be a member of the list, otherwise, the changing is rejected.

This action has the following format:

```
targetComputer action=ModifyInstallParameter
    name=computer_name
    paramName=parameter_name
    {default|paramValue=parameter_value}
```

default

Drops user settings of the specified parameter and sets it to default.

name

Specifies the name of the computer of which the install parameters will be modified.

paramName

Specifies the name of the parameter to be modified.

Note: The parameter “HostName” can only be modified as long as the specified target system has not been registered from the network.
If the system has already been registered, error CMD00149 is reported: “The target has already registered. The operation is not allowed.”

paramValue

Specifies the new value of the parameter.

If the parameter is a map list or extended map list, use the format "<key> <comment>"

The key and the comment are separated by a blank, and the comment is optional.

If the key contains blanks, enclose it in quotes.

Example: For an XP OS image change the setting of the "TimeZone" parameter to 095 or "095 Central Europe"

For a Vista image change the setting of the "TimeZone" parameter to "\"W. Europe Standard Time\"" or "\"W. Europe Standard Time\" UK time".

modifyOS—Assign OS Image to Target

With the modifyOS action you assign a new OS image to OSIM-managed or -un-managed systems.

The syntax of the action is as follows:

```
targetComputer  action= modifyOS
    name=target_name
    osImage=OS_image
    [bootserver]
    [macAddress=«Mac_address»]
```

name

Name of the target computer to which the OS image will be assigned.

osImage

Name of the OS image which is to be installed at the target

bootserver

Name of a boot server.

If the name is specified and the addressed target is not yet assigned to a boot server then the target is assigned to the specified boot server. If the addressed target is already assigned to a boot server and the name specified with the parameter then the action is only processed when the specified name matches the name of the assigned boot server. If the boot Server parameter is not coded then no assignments or checks as described above are performed.

macAddress

MAC address of the target.

If the system is not yet OSIM registered and does not have any MAC address assigned yet then this MAC address is taken to setup the system in OSIM.

If a MAC address is available in OSIM then it has to match the specified value otherwise the command returns an error.

If this argument is not coded then the addressed system is assumed to be OSIM registered already.

reactivateOS—Reactivate Stopped / Failed OS Installation Order

This action reschedules an installation for execution, if the target computer is OSIM managed and if a stopped or failed installation order request exists for it.

In any other case, the action will fail.

This action has the following format:

```
targetComputer action=reactivateOS
    name=computer_name
    [attime="YYYY-MM-DD hh:mm" ]
    [wakeup[={y|n=}]]
    [restart[={y|n}]]
    [waitBs[={y|n}]]
    [waitIm[={y|n}]]
```

attime

Specifies the Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

name

Specifies the name of the computer for which a stopped or failed installation is scheduled for execution again.

restart

This optional parameter specifies if a running system will be restarted when the associated installation request will be carried out.

If restart or restart=y is coded, then a restart will be performed.

If restart=n is coded, then no automatic restart will be performed and it is up to the user to reboot the system to start the OS installation process.

If the parameter is not coded, the default is given by restart=n.

waitBs

If waitBs or waitBs=y is coded then the activation request might be deferred until the addressed target is assigned a boot server. If it waitBs=n is coded then the request is not deferred because of a missing boot server assignment. The default is waitBs=n

waitIm

If waitIm or waitIm=y is coded then the activation request might be deferred until the associated boot and OS images are staged at the related boot server. If it waitIm=n is coded then the request is not deferred because of not yet staged images. The default is waitIm=n.

wakeup

This optional parameter specifies whether a target system in power standby mode shall be waked up for this installation or not.
If wakeup or wakeup=y is coded, then the systems will be waked up.
If wakeup=n is specified, no wake up will be performed.
If the parameter is not coded, the default wakeup=n will be used.

Example:

Assume that a stopped or failed OS installation should be scheduled for target computer dorabella. A wakeup should be performed, if the target system were in power standby mode.

Furthermore, an automatic restart should take place to start the installation process at the target system. The following command provides this request:

```
cadsmcmd targetcomputer action=reactivateOS name=dorabella wakeup restart
```

reinstallOS—Re-install Operating System

This action re-installs a target computer after crash.

If the system is managed by OSIM and if there is a current OS installation but no scheduled installation request regardless if it is active, stopped, or failed, then the reinstallation of the current OS is scheduled. In any other case, the request will fail.

This action has the following format:

```
targetComputer action=reInstallOS
    name=computer_name
    [atTime="YYYY-MM-DD hh:mm"]
    [wakeup[={y|n=}]]
    [restart[={y|n}]]
    [waitBs[={y|n}]]
    [waitIm[={y|n}]]
```

attime

Specifies the date and time when the action will be started.

The date has the ISO format "YYYY-MM-DD hh:mm".

name

Specifies the name of the computer to be reinstalled.

restart

This optional parameter specifies if a running system will be restarted when the associated installation request will be carried out.

If restart or restart=y is coded, then a restart will be performed.

If restart=n is coded, then no automatic restart will be performed and it is up to the user to reboot the system to start the OS installation process.

If the parameter is not coded, the default is given by restart=n.

waitBs

If waitBs or waitBs=y is coded then the activation request might be deferred until the addressed target is assigned a boot server. If it waitBs=n is coded then the request is not deferred because of a missing boot server assignment. The default is waitBs=n

waitIm

If waitIm or waitIm=y is coded then the activation request might be deferred until the associated boot and OS images are staged at the related boot server. If it waitIm=n is coded then the request is not deferred because of not yet staged images. The default is waitIm=n.

wakeup

This optional parameter specifies whether a target system in power standby mode shall be waked up for this installation or not.
If wakeup or wakeup=y is coded, then the systems will be waked up.
If wakeup=n is specified, no wake up will be performed.
If the parameter is not coded, the default wakeup=n will be used.

Example:

Assume that a crash has taken place and a reinstallation of the target computer dorabella should take place. If the target system were in power standby mode, the system should be waked up. Furthermore an automatic restart should be performed to start the installation process at the target system.

The following command provides this request:

```
cadsmcmd targetcomputer action=reinstallOS name=dorabella wakeup restart
```

removeFromOsim—Remove Targets from OSIM

This action removes the specified target objects from the OSIM administration as long as there is no scheduled configuration active.

The syntax of the action is as follows:

```
targetComputer  action= removeFromOSIM
    {{computer=target_name} |
    {macAddresses=Mac_address}}
```

computer

Name of an OSIM managed or unmanaged target to be removed from the OSIM administration. This option can be coded more than once for removing a list of OSIM objects.

macAddresses

Mac address of systems to be removed that are registered at OSIM but not at DSM. Such systems have no name, they only appear with their Mac address in OSIM.

Note : action=removeFromOSIM does not mean that the target is deleted from DSM. It is deleted from the OSIM only and the rest of DSM remains unchanged. The removal might only be a temporary OSIM clean up. For example, the removed system might re-register at OSIM with the next reboot.

showInstallParameter—Show Install Parameters of OS Installation Orders

This action lists for a target computer the install parameter related to the associated OS installation order.

It lists the installation parameters related to the OS installation order specified. If "planned" is coded the parameters of the planned OS installation order are listed, if "scheduled" is coded the parameters of the OS installation order that is being launched for installation are listed, and if none of them or "current" is coded they are listed from the current OS installation order.

If the target computer is not a OSIM managed system, the action fails.

This action has the following format:

```
targetComputer action=ShowInstallParameter
    name=computer_name
    [planningLevel={current|scheduled|planned}]
```

name

Specifies the name of the computer of which the install parameters are listed.

planningLevel

Optional Parameter. Determines the configuration from where the parameters are taken.

current

The install parameters of the current OS installation order are listed. (Default)

scheduled

The install parameters of the OS installation order scheduled for execution are listed.

planned

The install parameters of the planned OS installation order are listed.

setupOS (Obsolete Action)

With the enhancement of the action modifyOS, the action setupOS becomes obsolete. It is still accepted and processed by the CADSMCMD in batch, call, or pipe mode for compatibility reasons but will no longer be exposed in verbose mode.

templategroup—Software Template Commands

This section contains the following topics:

General Template Group Management

This section contains the following topics:

create—Create a Template Group / Software Template

This command creates a new software template.

This command has the following format:

```
templategroup action=create
    name=template_group_name
    [supergroup=group_name]
    [swTemplate=software_template_name]
    [comment=comment]
    [{query="query" | queryfile=file_name }]
    [engineName=engine_name
    [enginePeriod=engine_evalution_frequence]]
    [{transaction
    [norollback]|
```

```
synchronized }
[cascade[={y|n}]}]
[{setup_jobs | setup&activate_jobs }]
| [regenerate [ = {y/n}]]
| [{adhoc_only | adhoc_always}]
[evaluationstart ="YYYY-MM-DD hh:mm"]
[evaluationperiod=hours]
[calendarname=calendar_name]
[globaltime]
[jcPriority={1,...,10}]
[userMessage]
```

{adhoc_only | adhoc_always}

The coding has the following effects:

adhoc_only

Adhoc evaluation is enabled only when new or reinstalled targets report their inventory.

adhoc_always

Adhoc evaluation is enabled always when targets report their inventory.

If neither adhoc_only, nor adhoc_always are coded, then no adhoc handling is assumed.

calendarName

Specifies the name of the calendar that controls the time when the evaluation can take place.

Note: To activate a calendarname entry you also have to code "evaluationPeriod".

cascade

If "cascade" or "cascade=y" is coded, the job linkage of the container is switched to resolution of dependencies into install cascades.
If "cascade=n" is coded, the job linkage is switched to no resolution.

This parameter should only be coded, if the job linkage of the container is in batch processing or in synchronized mode, or if it switched into this mode by This action (by coding either "transaction" or "synchronized").

comment

Specifies a comment that is stored with the software template.

engineName

Specifies the name of an engine that will process the group evaluation.

If not coded any engine can process the evaluation.

This parameter is only valid if queryName or sdQuery is coded.

enginePeriod

Specifies in minutes the period in which the engine will re-evaluate the group. If not coded 1 min is assumed.If engineName is coded but not enginePeriod, a default of 1 minute is assumed.

This parameter is only valid if engineName is coded.

evaluationStart

Time, when the evaluation should start. The time should be coded in ISO format "YYYY-MM-DD hh:mm". If no EvaluationStart is coded, the actual time is used.

This parameter is to be used for query or template groups only.

evaluationPeriod

Period in which the query is evaluated. The period is specified in hours.

This parameter is usable for query or template groups only.

If it is omitted or if 0 is coded, no reevaluation is performed.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time. (default)

jcPriority

Specifies the priority of the job container.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded priority 5 is assumed.

name

Specifies the template group name.

If the group does not exist, it is created according to the specified group attributes. If it exists, the specified groups attributes are ignored and the specified software template is created and assigned to this group.

norollback

Specifies that no rollback of previous jobs of the same transaction be performed in case of failure.

query

Indicates that the template is also a query group. The query coded has to be a USD 4.0 SP1 legacy query.

queryfile

Specifies the the file in which the query is stored. The size of the query stored is only limited by the capacity of the server. The query has to be a legal SD 4.0 SP1 legacy query.

The query coded has to be a SD 4.0 SP1 legacy query.

regenerate

If "regenerate" or "regenerate=y" is coded the regeneration of jobs that have failed earlier for this template will be done.

If "regenerate=n" is coded it will not be done.

If the parameter is not coded, "regenerate=y" is assumed.

setup_jobs

According to the conformance check, set up jobs but do not activate.

If neither setup_jobs nor setup&activate_jobs is coded, then only a conformance check is assumed.

setup&activate_jobs

Set up and activate jobs according to the conformance check.

If neither setup_jobs nor setup&activate_jobs is coded, then only a conformance check is assumed.

Notes: If the parameter setup&activate_jobs is used from the command prompt or shell level, the parameter should be enclosed in quotes due to the special meaning of "&" to the command prompt or the shell. If you use batch scripts, you do not need quotes for this parameter.

supergroup

Specifies the name of an existing computer group where the group in question will be created as a subgroup. If not coded the group will be created at the system folder "Computers and Users".

swTemplate

Specifies the name of the software template the order will be assigned to.

If not coded the order will be assigned to a software template of default name, that is, “<group name> [$date $time]”.

synchronized

The job linkage option is set to "Synchronized job execution".

transaction

The job linkage option is set to "Batch job execution".

userMessage

Specifies the custom administrator message for a template group.

Example:

Suppose a software template is needed that meets the following requirements:

- The name of the group is "tgg"
- The jobs at the target should be executed in sequence. If a job fails then subsequent jobs will not be executed and already successful executed preceding jobs will be rolled back.
- Dependencies between packages should be resolved, that may result in cascades of install orders.
- In case of a conformance violation the software template generates jobs to resolve this conformance violation and schedules the jobs for execution.
- The software template addresses only those target systems that are belonging to the subnet 172.16.213.* and are running an operating system Windows NT Intel, Windows 2000 Intel, or Windows XP Intel.
- The evaluation of the group should start when the group is sealed. The evaluation will be repeated every 48 hours. The evaluation is driven by a calendar named tgCal.

The following command will create this group:

```
cadsmcmd templateGroup action=create name=tgg transaction cascade
"setup&activate_jobs" query="Target.Address MATCHES '172.16.213.*' AND
(Target.OS='Win NT Intel' OR Target.OS='Win 2000 Intel' OR Target.OS='Win XP Intel')"
evaluationPeriod=48 calendarName=tgCal
```

To control the actual list of software template , the following command lists all software template defined at the server:

```
cadsmcmd templateGroup action=list
```

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM
```

```
--------------------------------------------------------------------------------
List of template groups
--------------------------------------------------------------------------------
TG_KKKKK01A (local, query)
tgg (local, query)
tgs (local)
tg (local)
--------------------------------------------------------------------------------
Number of software templates found: 4
SDCMD<A000000>: OK
```

For controlling the attributes of the created software template the following command will return the attribute settings of the group:

```
cadsmcmd templateGroup action=showAttr name=tgg
```

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM
```

```
-------------------------------------------------------------------------------
Show attributes of template group "tgg"
-------------------------------------------------------------------------------
Name of template.group.................: tgg
Group type.............................: Query/Template group
Scope..................................: Local
Query..................................: USD_tgg [04/09/05 13:53:49.916]
Comment................................:
Engine name............................: <All Engines>
Engine Period (minutes)................: 1
Creation date..........................: 2005-04-09
Creation time..........................: 13:53
Change date............................: 2005-04-09
Change time............................: 13:53
Name of template.......................: tgg [$date $time]
..Scope................................: Local
..State................................: Diabled
..Job container priority...............: 5
..Sealed...............................: No
..Comment..............................:
..Properties of the template...........: 148
.......................................Transaction
.......................................Cascade install
.......................................Job setup and activation in case of nonc
.......................................+ onformity
Calendar name..........................: tgCal
Evaluation period......................: 48
Next evaluation date...................: 2005-04-09
Next evaluation time...................: 13:32
Last evaluation date...................: 2005-04-09
Last evaluation time...................: 13:53
Evaluation type mask...................: 196609
.......................................Periodical
.......................................Query evaluation
.......................................Template evaluation

SDCMD<A000000>: OK
```

delete—Delete software template

This action deletes an existing software template or the whole template group. The group must be unsealed!

This action has the following format:

```
templategroup action=delete
    name=template_group_name
    [swTemplate=software_template_name]
```

name

Specifies the template group name.

Whether the group is deleted or only a specified software template depends on the parameter swTemplate.

swTemplate

Specifies the name of the software template.

If specified, only the specified software template is deleted, the group itself and possible additional software templates associated with the group remain.

If not specified then the group and all its associated software templates are deleted.

deregister—Deregister Template

This action deregisters a software template or policy at the domain manager from the enterprise manager.

This action is valid on enterprise managers only.

This action has the following format:

```
templateGroup action=deregister
    name=template_group_name
    [swTemplate=software_template_name]
    cname=[name]
    {{area=area_name} | {domain=domain_name} | toAllAreas}
    [sendTime="YYYY-MM-DD hh:mm"]
    [haltTime="YYYY-MM-DD hh:mm"]
    [distPriority={1,...,10}]
```

area

Specifies the name of a domain manager from where the template is deregistered. The parameter can be coded more than once to address a list of domain managers.

This parameter should not be coded with the domain or the toAllAreas parameter.

cname

Specifies the optional unique name of a distribution container used to ship the deregistration request.

If a job container / distribution with the specified name already exists an error will be reported.
If "cname" is not coded, a generic name is generated.

distPriority

Specifies the priority that will be assigned to the distribution.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded priority 5 is assumed.

domain

Specifies the name of a domain manager group from its members the template is deregistered. The parameter can be coded more than once to address a list of domain manager groups.

This parameter should not be coded with the area or the toAllAreas parameter.

halttime

Specifies the date and time when the deregistration will be stopped, if it is not processed yet. The date is given in ISO format and should be at least 1 hour after the start time. If not ocded a halttime is generated as starttime + 1 hour.

name

Specifies the template group name.

sendtime

Specifies the start time of the distribution from the enterprise. The date is given in ISO format. If not coded the actual date/time is taken.

swTemplate

Specifies the name of the software template to be deregisterd.

If not coded the default name is taken, i.e. “<group name> [$date $time]”.

toAllAreas

All domain managers currently registered at the enterprise manager will be addressed for deregistering the specified software template or policy.

evaluate—Evaluate Template

This action evaluates a sealed software template of a group.

This action is valid only for domain managers

This action has the following format:

```
templateGroup action=evaluate
    name=template_group_name
    [swTemplate=software_template_name]
```

name

Specifies the template group name.

swTemplate

Specifies the name of the software template to be evaluated.

If not coded the order will be assigned to a software template of default name, that is, “<group name> [$date $time]”.

list—List Template Groups

This action lists the available template groups.

This action has the following format:

```
templategroup action=list
```

The following information is shown:

```
-------------------------------------
List of template groups
-------------------------------------
tg1 (Local)
tgq1 (Local, query)
tgqp (Local, query)
```

Type of scope and query groups are shown in parentheses.

Example

```
cadsmcmd templateGroup action=listComp name=tg


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
List of target computer assigned to template group "tgs"
--------------------------------------------------------------------------------
PPAPPGP0C
KKKKK01A
ORIONXH
SPICAYI
WWWWW01B
--------------------------------------------------------------------------------
Number of target computer read: 5
SDCMD<A000000>: OK
```

listExceptions—List Exceptions

This action lists all target computers which are a member of this software template but do not conform to the job assignment, that is, not all assigned jobs have been successfully completed yet ("violators").

This action is valid only for domain managers

This action has the following format:

```
templategroup action=listExceptions
    name=template_group_name
    [swTemplate=software_template_name]
```

name

Specifies the template group name of which the exceptions target members (violators) will be listed.

swTemplate

Specifies the name of the software template for which the exceptions will be listed.

If not coded the default name “<group name> [$date $time]” is taken.

The following information is provided for the target list:

```
------------------------------------------------------------------------
List of exception members of template group "<software template name>"
------------------------------------------------------------------------
<computer name>
<computer name>
<computer name>
```

Example:

Suppose information about those targets is needed, that are not conformant with the requirements of the software template tg . The following command lists all those targets currently assigned to tg but not conformant.

```
cadsmcmd templateGroup action=listExceptions name=tg
```

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM
```

```
--------------------------------------------------------------------------------
List of exception members of software template  "tg"
--------------------------------------------------------------------------------
GS_ELSEN
GS_MARIENLOH
GS_SALZKOTTEN
GS_WEWELSBURG
GS_WEWER

SDCMD<A000000>: OK
```

listMembership—List Membership of the Template Group

This action lists all computer groups that contain the specified software template as a subgroup.

This action has the following format:

```
templategroup action=listMembership
    name=template_group_name
```

name

Specifies the template group name of which the memberships will be listed.

The following information is provided:

```
--------------------------------------------------------------------------------
"<software template name >" is member of the following groups
--------------------------------------------------------------------------------
<group name> (<group type>, <group state>, <registration date>)
<group name> (<group type>, <group state>, <registration date>)
<group name> (<group type>, <group state>, <registration date>)
```

Example:

Suppose information about those groups is needed that contain the software template tg as a subgroup. The following command lists all the groups containing tg:

```
cadsmcmd templateGroup action=listMembership name=tg



CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
"tg" is member of the following Groups
--------------------------------------------------------------------------------
cg (Group, Operational, 2005-04-08 12:08)
superTg (Template group, Conformant, 2005-04-08 12:00)

SDCMD<A000000>: OK
```

modify—Modify Attributes of a Template Group / Software Template

This command modifies the attributes of an existing software group and template.

This command has the following format:

```
templategroup action=modify name=template_group_name
    [newName=template_group_name]
    [swTemplate=software_template_name]
    [newSwTemplate=new_software_template_name]
    [comment=comment]
    [{query="query" | queryfile=filename | noquery}]
    [engineName=engine_name
    [enginePeriod=engine_evalution_frequence]]
    [{transaction|synchronized|nolinkage}]
```

```
[rollback[={y|n}]]
[cascade[={y|n}]]
[{nosetup | setup_jobs | setup&activate_jobs}]
| [regenerate [ = {y/n}]]
| [{adhoc_disable | adhoc_only | adhoc_always}]
[evaluationstart="YYYY-MM-DD hh:mm"]
[evaluationperiod=hours]
[calendarname=calendar_name]
[globalTime[={y|n}]]
[jcPriority={1,...,10}]
[userMessage]
```

{adhoc_disable | adhoc_only | adhoc_always}

The coding has the following effects:

adhoc_disabled

Adhoc evaluation is disabled.

adhoc_only

Adhoc evaluation is enabled only when new or reinstalled targets report their inventory.

adhoc_always

Adhoc evaluation is enabled always when targets report their inventory.

calendarname

Specifies the name of the calendar that controls the time when the evaluation can take place.

Note: To activate a calendarname entry you also have to code "evaluationPeriod".

cascade

If "cascade" or "cascade=y" is coded, the job linkage of the container is switched to resolution of dependencies into install cascades.
If "cascade=n" is coded, the job linkage is switched to no resolution.

This parameter should only be coded, if the job linkage of the container is in batch processing or in synchronous mode, or switched to it by this command (coding either "transaction" or "synchronous").

comment

a comment will be replaced by the specified value.

Note: To erase an existing comment, enter the empty string ("").

engineName

Specifies the name of an engine that will process the group evaluation.

If an empty string is coded then any engine can process the query

If currently no query is assigned to the group a warning is given and the parameter is ignored.

enginePeriod

Specifies in minutes the period in which the engine will re-evaluate the group. If not coded 1 min is assumed.If currently no query is assigned to the group and no special engine to process the evaluation then a warning is given and the parameter is ignored.

evaluationstart

Time, when the evaluation should start. The time should be coded in ISO format "YYYY-MM-DD hh:mm". If no EvaluationStart is coded, the actual time is used.

This parameter is to be used for query or template groups only.

evaluationperiod

Period in which the query is evaluated. The period is specified in hours.

This parameter is usable for query or template groups only.

If 0 (zero) is entered, the periodical evaluation is switched off.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

jcPriority

Specifies the new priority that will be assigned to the job container.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded the priority is not changed.

name

Specifies the template group name to be modified.

newName

Specifies the new template group name.

newSwTemplate

Specifies the new name of the software template in question.

nolinkage

Run the job independently of the other jobs in the container.

noquery

A possibly coded query is deleted.

nosetup

If this parameter is coded only conformance checks but no job setups will be performed.

query

Indicates that the template is also a query group. The query coded has to be a USD 4.0 SP1 legacy query.

queryfile

Specifies the file in which the query is stored. The size of the query stored is only limited by the capacity of the server. The query has to be a legal SD 4.0 SP1 legacy query.

regenerate

If "regenerate" or "regenerate=y" is coded the regeneration of jobs that have failed earlier for this template will be done.

If "regenerate=n" is coded it will not be done.

If the parameter is not coded, "regenerate=y" is assumed.

rollback

Set the job linkage option to "Enable Transaction".

This parameter should only be coded, if the container is already in the batch processing mode or if it is switched to it by this command (parameter "transaction").

setup_jobs

According to the conformance check, set up jobs but do not activate.

setup&activate_jobs

Set up and activate jobs according to the conformance check.

Notes: If the parameter setup&activate_jobs is used from the command prompt or shell level, the parameter should be enclosed in quotes due to the special meaning of "&" to the command prompt or the shell. If you use batch scripts, you do not need quotes for this parameter.

swTemplate

Specifies the name of the software template for which the attributes will be modified.

If not coded the default name “<group name> [$date $time]” is taken.

synchronized

The job linkage option is set to "Synchronized job execution".

transaction

The job linkage option is set to "Batch job execution".

userMessage

Modifies the custom administrator message while updating a template group.

Note: If a parameter is not coded the related attribute is not changed.

Example:

Suppose a software template named "tg" is defined on a domain manager. This group is considered changed in the following way:

- The linkage convention should be set to batch job processing with rollback and automatically cascade installation resolution.
- The conformance check should result in the jobs to be activated automatically.
- The software template should be evaluated every 12 hours starting on October 1, 2005, at 01:00 am.
- The time specifications are to be interpreted on enterprise manager time base.

The following command modifies the software template as required:

```
cadsmcmd templateGroup action=modify name=tg transaction rollback cascade
"setup&activate_jobs" evaluationstart="2005-10-01 01:00" evaluationperiod=12
GlobalTime
```

register—Register Template

This action registers a software template or policy at the domain manager from the enterprise manager.

This action is valid on enterprise managers only.

This action has the following format:

```
templateGroup action=register
    name=template_group_name
    [swTemplate=software_template_name]
    cname=[name]
    {{area=area_name} | {domain=domain_name} | toAllAreas}
    [sendTime="YYYY-MM-DD hh:mm"]
    [haltTime="YYYY-MM-DD hh:mm"]
    [distPriority={1,...,10}]
```

area

Specifies the name of a domain manager the template is registered at. The parameter can be coded more than once to address a list of domain managers.

This parameter should not be coded with the domain or the toAllAreas parameter.

cname

Specifies the optional unique name of a distribution container used to ship the registration request.

If a job container / distribution with the specified name already exists an error will be reported.
If "cname" is not coded, a generic name is generated.

distPriority

Specifies the priority that will be assigned to the distribution.

The priority specified has to be numeric and its value has to be greater or equal 1 and less or equal 10.

If not coded priority 5 is assumed.

domain

Specifies the name of a domain manager the template is registered at its members. The parameter can be coded more than once to address a list of domain manager groups.

This parameter should not be coded with the area or the toAllAreas parameter.

halttime

Specifies the date and time when the registration will be stopped, if it is not processed yet. The date is given in ISO format and should be at least 1 hour after the start time. If not coded a halttime is generated as sendtime + 1 hour.

name

Specifies the template group name.

sendtime

Specifies the start time of the distribution from the enterprise. The date is given in ISO format. If not coded the actual date/time is taken.

swTemplate

Specifies the name of the software template to be registerd.

If not coded the default name is taken, i.e.“<group name> [$date $time]”.

toAllAreas

All domain managers currently registered at the enterprise manager will be addressed for registering the specified software template or policy.

showAttr—Show Attributes of software template

This action shows the attributes of the specified software template.

This action has the following format:

```
templategroup action=showattr
    name=template_group_name
```

name

Specifies the template group name for which the attributes and those of the associated software templates will be shown.

seal—Seal a Software Template

This action seals a software template.

This action is valid only for domain managers

This action has the following format:

```
templategroup action=seal
    name=template_group_name
    [swTemplate=software_template_name]
```

name

Specifies the template group name.

swTemplate

Specifies the name of the software template to be sealed.

If not coded the default name “<group name> [$date $time]” is taken.

setupExecuteJobs—Setup and Execute Jobs for Software Template

This action sets up and activates the jobs for the software template.

This action is valid only for domain managers

This action has the following format:

```
templategroup action=setupexecutejobs
    name=template_group_name
    [swTemplate=software_template_name]
```

name

Specifies the template group name.

swTemplate

Specifies the name of the software template for which the jobs will be setup and activated.

If not coded the default name “<group name> [$date $time]” is taken.

setupJobs—Setup Jobs for Software Template

This action sets up the jobs for the software template but does not activate them.

This action is valid only for domain managers

This action has the following format:

```
templategroup action=setupjobs
    name=template_group_name
    [swTemplate=software_template_name]
```

name

Specifies the template group name.

swTemplate

Specifies the name of the software template for which the jobs will be setup.

If not coded the default name “<group name> [$date $time]” is taken.

unseal—Unseal a Software Template

This action unseals a software template.

This action is valid only for domain managers

This action has the following format:

```
templategroup action=unseal
    name=template_group_name
    [swTemplate=software_template_name]
```

name

Specifies the template group name.

swTemplate

Specifies the name of the software template to be unsealed.

If not coded the default name “<group name> [$date $time]” is taken.

Template Groups: Target Computer Management

This section contains the following topics:

addComp—Add Computer List to software template

This action adds a list of computers to the software template. The software template has to be unsealed and the software template should not be a query group.

This action is valid only for domain managers

This action has the following format:

```
templategroup action=addcomp
    name=template_group_name
    {computer=computer_name}
```

name

Specifies the template group name to be expanded is given.

computer

Specifies the name of the computer to be added to the software template.

The parameter can be coded several, at least one times.

Example:

Assume a software template tgs that is not associated with a query. To determine the targets the user has to add and remove, the target computer manually while the group is unsealed. Suppose the group is sealed and containing a target HHHHH01B that should be removed and a new target named ORIONXH and SPICAYI should be added. After these changes, the group should be sealed again.

First, a list of all targets assigned to the group is provided by the following command:

```
cadsmcmd templateGroup action=listComp name=tgs


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off
```

```
Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM

--------------------------------------------------------------------------------
List of target computer assigned to template group "tgs"
--------------------------------------------------------------------------------
PPAPPGP0C
HHHHH01B
KKKKK01A
WWWWW01B
--------------------------------------------------------------------------------
Number of target computer read: 4
SDCMD<A000000>: OK
```

Now to modify the list of assigned targets the group has to be unsealed.

```
cadsmcmd templateGroup action=unseal name=tgs
```

Next, the HHHHH01B will be removed from the list by the following command:

```
cadsmcmd templateGroup action=removeComp name=tgs computer=HHHHH01B
```

The next command will add ORIONXH and SPICAYI to the list:

```
cadsmcmd templateGroup action=addComp name=tgs computer=ORIONXH computer=SPICAYI
```

The modified list is shown by the following command:

```
cadsmcmd templateGroup action=listComp name=tgs


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Connecting to manager "<default manager>" as user "<default user>" ...OK.

Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CONF USD OSIM
```

```
--------------------------------------------------------------------------------
List of target computer assigned to template group "tgs"
--------------------------------------------------------------------------------
PPAPPGP0C
KKKKK01A
ORIONXH
SPICAYI
WWWWW01B
--------------------------------------------------------------------------------
Number of target computer read: 6
SDCMD<A000000>: OK
```

At last, the group has to be sealed by the following command:

```
sdcmd templateGroup action=seal name=tgs
```

listComp—List Computers of a Template Group

This action lists the computers assigned to the specified software template.

This action is valid only for domain managers

This action has the following format:

```
templategroup action=listComp
    name=template_group_name
```

name

Specifies the template group name.

removeComp—Remove Computer from Software Template Group

This action removes a list of computers from the specified software template. The software template has to be unsealed.

This action is valid only for domain managers

This action has the following format:

```
templategroup action=removecomp
    name=template_group_name
    {computer=computer_name }
```

computer

Specifies the name of the computer to be removed from the software template.

The parameter can be coded more than once.

name

Specifies the name of the template group name.

Example:

The computer HHHHH01B will be removed from the software template tgs by the following command:

```
cadsmcmd templateGroup action=removeComp name=tgs computer=HHHHH01B
```

Template Groups: Item Management

This section contains the following topics:

activateItem—Activate Item in software template

This command assigns an *activate* job to the software template.

This command has the following format:

```
templategroup action=activateItem
    name=template_group_name
    [swTemplate=software_template_name]
    item=item_name
    version=item_version
    procedure=procedure_name
    installedWith=install_procedure_name
    [repeat]
    after={exacttime|boottime}
    [calendarname=delivery_calendar]
```

```
[{preaction={none|reboot|logoff}|bootbefore}]
[{postaction=
    {none|reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}
    |bootafter}]
[promptUser[={y|n}]]
[allowCancel[={y|n}]]
[execTimedOut[={y|n}]]
[prompt=days.hours]
[offline[={y|n}]]
[runAtShutdown[={y|n}]]
[preventLogon[={y|n}]]
[globalTime]
[nocalendar]
[resolveQuery]
[stagingServer]
[parameters=user_parameters]
[{jobTimeout=d.h|timeout=hours}]
[userJobMessage]
```

after

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

allowCancel

If "allowCancel" or "allowCancel=y" is coded, the user will be granted to cancel the job's execution.
If "allowCancel=n" is coded, the user will not be granted.

If the parameter is not coded, the default is given by the related procedure's job option value.

calendarname

Specifies the name of the calendar that controls the time when the evaluation can take place.

execTimedOut

If "execTimedOut " or "execTimedOut=y" is coded, the job's execution will be automatically started when the user prompt times out.
If "execTimedOut=n", the job will not automatically be started.

If the parameter is not coded, the default is given by the related procedure's job option value.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

installedWith

Specifies the name of the install procedure that has installed the item at the target system.

item

Specifies the name of the item to be activated.

name

Specifies the template group name to which the activate order is added.

jobTimeout

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

noCalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

(default)

offline

If "offline" or "offline=y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.
If "offline=n" the connection will not be released.

parameters

Specifies the user parameters for the procedure.
Multiple user parameters must be separated by spaces inside double quotes.
If an empty string is coded, then no parameters are assumed.

If the parameter is not coded, then the default user parameters are given by the procedure definition.

postaction

Specifies the necessary actions that must take place after completion of the specified procedure. The following values are valid:

none

Performs no post-action.

reboot

Restarts the system after completion of the procedure.

logoff

Logs off the user after completion of the procedure.

rebootAtEnd

Restarts the system after completion of all the jobs of this container.

logoffAtEnd

Logs off the user after completion of all jobs of this container.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

preaction

Specifies that necessary actions must take place before the specified procedure starts. The following values are valid:

none

Performs no pre-action.

reboot

Restarts the system before the start of the procedure.

logoff

Logs off the user before the start of the procedure.

preventLogon

If "preventLogon" or "preventLogon=y" is coded, then the user logon will be rejected during the job's runtime. If a user is already logged on, the execution of the job is delayed until the user logs off.
If "preventLogon=n" the user logon will not be rejected.

If the parameter is not coded the default is given by the related procedure's job option value.

procedure

Specifies the name of a procedure (type: activate) to be run.

prompt=d.h

Specifies the time period in which the user will be prompted for the job's start.

The period's format is "*d.h*" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0). If the specified value falls below the valid minimum, the value is replaced by the minimum. If the specified value exceeds the valid maximum, then the value is replaced by the maximum. No warning is given.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

If the parameter is not coded, the default is given by the related procedure's job option value.

repeat

If coded the job is repeated every time an evaluation takes place.

resolveQuery

Evaluate a possibly related query group before the job is generated.

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

swTemplate

Specifies the name of the software template the order will be assigned to.

If not coded the order will be assigned to a software template of default name, that is, “<group name> [$date $time]”.

timeout

Specifies the timeout specifies the expiration period of a job in the form “h” where h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
timeout should not be coded with the jobTimeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 168

version

Specifies the version of the item to be activated.

userJobMessage

Specifies the custom administrator message for an *activate* job in a template group.

The parameters "allowCancel," "execTimedOut" and "prompt" are valid, if the "promptUser" option is already set or is set by this command using the "promptUser" parameter.
If neither the "promptUser" option of a job nor the "promptUser" parameter of the command is set, these parameters will be ignored.
No warning is given.

Example:

Consider the product "prod" of version "1.0/00" and a software template "tg". And at this time an activation of the product should be assigned to the software template .The activating routine to be run is named "act" and should only run on those systems that have installed the product with the install routine "inst_2". The provided job should meet the following requirements:

- The activation of the targets should be started immediately.
- Calendars assigned to the targets will be ignored for the activation schedule.
- The user should be prompted for the jobs execution, but the user is not granted to cancel the job. The job is automatically started after the prompt has timed out. The prompt time out is given by the default.

The following command will assign the required job to the specified software template.

```
cadsmcmd templateGroup action=activateItem name=tg item=prod version=1.0/00
procedure=act installedWith=inst_2 after=exacttime nocalendar promptUser
execTimedOut
```

addItem—Add Item to a Software Template

This command assigns an *install* job to the software template.

This command has the following format:

```
templategroup action=addItem
    name=template_group_name
    [swTemplate=software_template_name]
    item=item_name
    version=item_version
    procedure=procedure_name
    after={exacttime|boottime}
    [calendarname=delivery_calendar]
    [{preaction={none|reboot|logoff}|bootbefore}]
    [{postaction=
        {none|reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}
        |bootafter}]
    [promptUser[={y|n}]]
    [allowCancel[={y|n}]]
    [execTimedOut[={y|n}]]
    [prompt=days.hours]
    [offline[={y|n}]]
    [runAtShutdown[={y|n}]]
    [preventLogon[={y|n}]]
    [globalTime]
    [nocalendar]
    [resolveQuery]
    [stagingServer]
    [parameters=user_parameters]
    [{jobTimeout=d.h|timeout=hours}]
    [reinstall[={y|n}]]
    [userJobMessage]
```

after

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

allowCancel

If "allowCancel" or "allowCancel=y" is coded, the user will be granted to cancel the job's execution.
If "allowCancel=n" is coded, the user will not be granted.

If the parameter is not coded, the default is given by the related procedure's job option value.

calendarname

Specifies the name of the calendar that controls the time when the evaluation can take place.

execTimedOut

If "execTimedOut " or "execTimedOut=y" is coded, the job's execution will be automatically started when the user prompt times out.
If "execTimedOut=n", the job will not automatically be started.

If the parameter is not coded, the default is given by the related procedure's job option value.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

item

Specifies the name of the item to be added.

jobTimeout

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

name

Specifies the template group name to which the install order is added.

nocalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

offline

If "offline" or "offline=y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.
If "offline=n" the connection will not be released.

parameters

Specifies the user parameters for the procedure.
Multiple user parameters must be separated by spaces inside double quotes.
If an empty string is coded, then no parameters are assumed.

If the parameter is not coded, then the default user parameters are given by the procedure definition.

postaction

Specifies the necessary actions that must take place after completion of the specified procedure. The following values are valid:

none

Performs no post-action.

reboot

Restarts the system after completion of the procedure.

logoff

Logs off the user after completion of the procedure.

rebootAtEnd

Restarts the system after completion of all the jobs of this container.

logoffAtEnd

Logs off the user after completion of all the jobs of this container.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

preaction

Specifies that necessary actions must take place before the specified procedure starts. The following values are valid:

none

Performs no pre-action.

reboot

Restarts the system before the start of the procedure.

logoff

Logs off the user before the start of the procedure.

preventLogon

If "preventLogon" or "preventLogon=y" is coded, then the user logon will be rejected during the job's runtime. If a user is already logged on, the execution of the job is delayed until the user logs off.
If "preventLogon=n" the user logon will not be rejected.

procedure

Specifies the name of the install procedure to be run.

prompt

Specifies the time period in which the user will be prompted for the job's start.

The period's format is "*d.h*" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0). If the specified value falls below the valid minimum, the value is replaced by the minimum. If the specified value exceeds the valid maximum, then the value is replaced by the maximum. No warning is given.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

reinstall

If if "reinstall" or reinstall=y, a job container to install the software package will be set up during every evaluation of the policy, independent of whether the policy has already previously been installed with the selected installation procedure.

Note: A job container will not be set up during the evaluation of the policy if another installation job with the selected installation procedure or an uninstallation job is in progress while the policy evaluation takes place.

resolveQuery

Evaluate a possibly related query group before the job is generated.

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

stagingServer

Deliver the related product also to the staging libraries of the scalability servers of the target computers.

This parameter applies to procedures of task type "install" only!

swTemplate

Specifies the name of the software template the order will be assigned to.

If not coded the order will be assigned to a software template of default name, that is, “<group name> [$date $time]”.

timeout

Specifies the timeout specifies the expiration period of a job in the form “h” where h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
timeout should not be coded with the jobTimeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 168

version

Specifies the version of the item to be added.

The parameters "allowCancel," "execTimedOut" and "prompt" are valid, if the "promptUser" option is already set or is set by this command using the "promptUser" parameter.
If neither the "promptUser" option of a job nor the "promptUser" parameter of the command is set, these parameters will be ignored.
No warning is given.

userJobMessage

Specifies the custom administrator message for an *install* job in a template group.

Example:

Assume a product "prod" of version "1.0/00" and a software template "tg". The product is considered as being assigned to the software template for installation by the install routine named "inst". The install order should meet the following requirements:

- The delivery of the package has to meet the time frames given by the calendar "DelCal".
- The installation on the targets should be started as soon as possible.
- Before the installation starts the user should be logged off.
- During the installation, any user logon is rejected.
- The installation runs offline from the server.

- Calendars assigned to the targets will be ignored for the installation schedule.
- After installation the system should be rebooted, but this can be delayed until all jobs of the related job container have been processed.
- The product has to be delivered to the software libraries of those scalability servers related to the addressed targets.
- The job will be timed out after 24 hours.

The following command will assign the required job to the specified software template.

```
cadsmcmd templateGroup action=addItem name=tg item=prod version=1.0/00
procedure=inst calendarname=DelCal after=exacttime preaction=logoff offline
PreventLogon nocalendar postaction=rebootAtEnd stagingServer JobTimeout=1.0
```

configureItem—Assign a Configure Job to a Software Template

This command assigns a *configure* job to the software template.

This command has the following format:

```
templategroup action=configureItem
    name=template_group_name
    [swTemplate=software_template_name]
    item=item_name
    version=item_version
    procedure=procedure_name
    installedWith=install_procedure_name
    [repeat]
    after={exacttime|boottime}
    [calendarname=delivery_calendar]
    [{preaction={none|reboot|logoff}|bootbefore}]
    [{postaction=
        {none|reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}
        |bootafter}]
    [promptUser[={y|n}]]
    [allowCancel[={y|n}]]
    [execTimedOut[={y|n}]]
    [prompt=days.hours]
    [offline[={y|n}]]
    [runAtShutdown[={y|n}]]
    [preventLogon[={y|n}]]
    [globalTime]
    [nocalendar]
    [resolveQuery]
    [parameters=user_parameters]
    [{jobTimeout=d.h|timeout=hours}]
    [userJobMessage]
```

after

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

allowCancel

If "allowCancel" or "allowCancel=y" is coded, the user will be granted to cancel the job's execution.
If "allowCancel=n" is coded, the user will not be granted.

If the parameter is not coded, the default is given by the related procedure's job option value.

calendarname

Specifies the name of the calendar that controls the time when the evaluation can take place.

execTimedOut

If "execTimedOut " or "execTimedOut=y" is coded, the job's execution will be automatically started when the user prompt times out.
If "execTimedOut=n", the job will not automatically be started.

If the parameter is not coded, the default is given by the related procedure's job option value.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

(default)

installedWith

Specifies the name of the install procedure that has installed the item at the target system.

item

Specifies the name of the item to be configured.

jobTimeout

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

name

Specifies the template group name to which the configure order is added.

noCalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

If "noCalendar=n" is coded, the calendar will not be ignored any longer.

(default)

offline

If "offline" or "offline=y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.
If "offline=n" the connection will not be released.

If the parameter is not coded, the default is given by the related procedure's job option value.

parameters

Specifies the user parameters for the procedure.
Multiple user parameters must be separated by spaces inside double quotes.
If an empty string is coded, then no parameters are assumed.

If the parameter is not coded, then the default user parameters are given by the procedure definition.

postaction

Specifies the necessary actions that must take place after completion of the specified procedure. The following values are valid:

none

Performs no post-action.

reboot

Restarts the system after completion of the procedure.

logoff

Logs off the user after completion of the procedure.

rebootAtEnd

Restarts the system after completion of all the jobs of this container.

logoffAtEnd

Logs off the users after completion of all the jobs of this container.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

preaction

Specifies that necessary actions must take place before the specified procedure starts. The following values are valid:

none

Performs no pre-action.

reboot

Restarts the system before the start of the procedure.

logoff

Logs off the user before the start of the procedure.

preventLogon

If "preventLogon" or "preventLogon=y" is coded, then the user logon will be rejected during the job's runtime. If a user is already logged on, the execution of the job is delayed until the user logs off.
If "preventLogon=n" the user logon will not be rejected.

procedure

Specifies the name of a procedure of type configure to be run.

prompt=d.h

Specifies the time period in which the user will be prompted for the job's start.

The period's format is "*d.h*" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0). If the specified value falls below the valid minimum, the value is replaced by the minimum. If the specified value exceeds the valid maximum, then the value is replaced by the maximum. No warning is given.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

repeat

If coded the job is repeated every time an evaluation takes place.

resolveQuery

Evaluate a possibly related query group before the job is generated.

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

swTemplate

Specifies the name of the software template the order will be assigned to.

If not coded the order will be assigned to a software template of default name, that is, "<group name> [$date $time]".

timeout

Specifies the timeout specifies the expiration period of a job in the form “h” where h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
timeout should not be coded with the jobTimeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 168

version

Specifies the version of the item to be configured.

userJobMessage

Specifies the custom administrator message for a *configure* job in a template group.

The parameters "allowCancel," "execTimedOut" and "prompt" are valid, if the "promptUser" option is already set or is set by this command using the "promptUser" parameter.
If neither the "promptUser" option of a job nor the "promptUser" parameter of the command is set, these parameters will be ignored.
No warning is given.

Example

Consider the product "prod" of version "1.0/00" and a software template "tg". At this time a configuration of the product should be assigned to the software template .The configuration routine to be run is named "conf" and should only run on those system that have installed the product with the install routine "inst_1". The provided job should meet the following requirements:

- The configuration on the targets should be started after the next reboot.
- During the installation, any user logon is rejected.
- Calendars assigned to the targets will be ignored for the configuration's schedule.
- After configuration the system should be rebooted immediately.
- The job will be timed out after 36 hours.

The following command will assign the required job to the specified software template.

```
cadsmcmd templateGroup action=configureItem name=tg item=prod version=1.0/00
procedure=conf installedWith=inst_1 after=boottime PreventLogon nocalendar
postaction=reboot JobTimeout=1.12
```

modifyItem—Modify Item in software template

This command modifies the attributes of a job assigned to a software template.

This command has the following format:

```
templategroup action=modifyItem name=template_group_name
    [swTemplate=software_template_name]
    jobName=name_of_the_item_order
    [newJobName=new_name_of_the_item_order]
    after={exacttime|boottime}
    [calendarname=delivery_calendar]
    [preaction={none|reboot|logoff}]
    [postaction={none|reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}]
    [promptUser[={y|n}]]
    [allowCancel[={y|n}]]
    [execTimedOut[={y|n}]]
    [prompt=days.hours]
    [offline[={y|n}]]
    [runAtShutdown[={y|n}]]
    [preventLogon[={y|n}]]
    [globalTime]
    [nocalendar]
    [resolveQuery]
    [parameters=user_parameters]
    [repeat={y|n}]
    [stagingserver]
    [jobTimeout=d.h]
    [reinstall[={y|n}]]
    [userJobMessage]
```

after

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

allowCancel

If "allowCancel" or "allowCancel=y" is coded, the user will be granted to cancel the job's execution.
If "allowCancel=n" is coded, the user will not be granted.

If the parameter is not coded, the default is given by the related procedure's job option value.

calendarname

Specifies the name of the calendar that controls the time when the evaluation can take place.

If an empty string ("") is coded, the calendarname will be removed.

execTimedOut

If "execTimedOut " or "execTimedOut=y" is coded, the job's execution will be automatically started when the user prompt times out.
If "execTimedOut=n", the job will not automatically be started.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time.

jobName

Specifies the name of the job to be modified.

jobTimeout

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

name

Specifies the template group name of which an item is modified.

newJobName

Specifies the new name of the job .

noCalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

offline

If "offline" or "offline=y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.
If "offline=n" the connection will not be released.

parameters

Specifies the user parameters for the procedure.
Multiple user parameters must be separated by spaces inside double quotes.
If an empty string is coded, then no parameters are assumed.

postaction

Specifies the necessary actions that must take place after completion of the specified procedure. The following values are valid:

none

Performs no post-action.

reboot

Restarts the system after completion of the procedure.

logoff

Logs off the user after completion of the procedure.

rebootAtEnd

Restarts the system after completion of all the jobs of this container.

logoffAtEnd

Logs off the user after completion of all the jobs of this container.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

preaction

Specifies that necessary actions must take place before the specified procedure starts. The following values are valid:

none

Performs no pre-action.

reboot

Restarts the system before the start of the procedure.

logoff

Logs off the user before the start of the procedure.

preventLogon

If "preventLogon" or "preventLogon=y" is coded, then the user logon will be rejected during the job's runtime. If a user is already logged on, the execution of the job is delayed until the user logs off.
If "preventLogon=n" the user logon will not be rejected.

prompt=d.h

Specifies the time period in which the user will be prompted for the job's start.

The period's format is "*d.h*" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0). If the specified value falls below the valid minimum, the value is replaced by the minimum. If the specified value exceeds the valid maximum, then the value is replaced by the maximum. No warning is given.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

reinstall

If if "reinstall" or reinstall=y, a job container to install the software package will be set up during every evaluation of the policy, independent of whether the policy has already previously been installed with the selected installation procedure.

Note: A job container will not be set up during the evaluation of the policy if another installation job with the selected installation procedure or an uninstallation job is in progress while the policy evaluation takes place.

repeat

If coded the job is repeated every time an evaluation takes place.

resolveQuery

Evaluate a possibly related query group before the job is generated.

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

stagingServer

Deliver the related product also to the staging libraries of the scalability servers of the target computers.

This parameter applies to procedures of task type "install" only!

swTemplate

Specifies the name of the software template where the item to be modified is located.

If not coded the default name “<group name> [$date $time]” is taken.

Note: If a parameter is not coded the related attribute is not changed.

userJobMessage

Modifies the custom administrator message of a job, while updating the job properties, in a template group.

The parameters "allowCancel," "execTimedOut" and "prompt" are valid, if the "promptUser" option is already set or is set by this command using the "promptUser" parameter.
If neither the "promptUser" option of a job nor the "promptUser" parameter of the command is set, these parameters will be ignored.
No warning is given.

Example:

The uninstallation job of the software template "tg" is considered as modified in the following way:

- User parameters "-x -Nc -ro" has to be added.
- The job time out is reduced to 12 hours.
- During uninstall any user logon should be rejected.
- The users have to be logged off before job start.
- The job has to resolve a specified group query before it is generated and launched.

The following command meets these requirements:

```
cadsmcmd templateGroup action=modifyItem

        name=tg

        jobName="prod 1.0/00:uninstall"

        preaction=logoff

        PreventLogon

        ResolveQuery

        parameters="-x -Nc -ro"

        JobTimeout=0.12
```

listItem—List Items of a Software Template

This action lists the items of the "Assigned Jobs" folder for the specified software template.

This action has the following format:

```
templategroup action=listItem
    name=template_group_name
    [swTemplate=software_template_name]
```

name

Specifies the name template group name.

swTemplate

Specifies the name of the software template of which the items will be listed.

If not coded the default name “<group name> [$date $time]” is taken.

The following information is shown:

```
------------------------------------------------
List of items assigned to template group "tg1"
------------------------------------------------
tstool_1 1.0:inst
tstool_1 1.0:conf
------------------------------------------------
```

Example:

Suppose information about all the items assigned to a software template tg is needed. The following command provides a list of all items assigned to the group:

```
cadsmcmd templateGroup action=listItem name=tg


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM

------------------------------------------------------------------------------
List of items assigned to template group "tg"
------------------------------------------------------------------------------
tstbase 1.0: inst <1>
tstbase 1.0: remdir <2>
tstbase 1.0: setdir <3>
------------------------------------------------------------------------------
Number of assigned items read: 3
SDCMD<A000000>: OK
```

The order number of the jobs is shown in angled brackets (<>).

For retrieving more detailed information about a certain job see showAttrItem (see definition on page 493).

Remove Item of software template

This action removes a job, identified by item, version, and procedure, from the software template.

This action has the following format:

```
templategroup action=removeitem
    name=template_group_name
    [swTemplate=software_template_name]
    item=item_name
    version=item_version
    procedure=procedure_name
```

item

Specifies the name of the item addressed by the job to be removed.

name

Specifies the template group name.

procedure

The name of the procedure specified in the job to be removed.

swTemplate

Specifies the name of the software template from which the item will be removed.

If not coded the default name “<group name> [$date $time]” is taken.

version

Specifies the version of the item addressed by the job to be removed.

Example:

Suppose the procedure "inst" of product "prod" of version "1.0/00" is assigned to the software template "tg" and should be removed now. The following command will remove the item:

```
cadsmcmd templateGroup action=removeItem name=tg item=prod version=1.0/00
procedure=install
```

repositionJobs—Reposition Jobs in software template

This action reorders the jobs (items) assigned to a software template by specifying a new order number for an item.

This action has the following format:

```
templategroup action=repositionJobs name=template_group_name
    [swTemplate=software_template_name]
    jobName=job_name
    position=new_position
```

name

Specifies the template group name of which the jobs are to be reordered.

jobName

Specifies the name of the job to be reordered.

position

Specifies the new position new position of the job after reordering.

The number to be specified has to be greater than 0 and less or equal to the amount of jobs assigned to the template.

swTemplate

Specifies the name of the software template where the specified job will be positioned.

If not coded the default name “<group name> [$date $time]” is taken.

Example:

Suppose the job "prod 1.0/00:inst" is assigned to the software template "tg" and shows order number 11. To ensure correct job execution, the job has to be placed in position 4. The following command will reposition the job:

```
cadsmcmd templateGroup action=repositionJobs name=tg jobName="prod 1.0/00:install"
position=4
```

The order numbers of those jobs that occupied the positions 4 to 10, before the repositioning, will be increased by one.

showAttrItem—Show Attributes of Item in software template

This action shows the attributes of a job assigned to a software template.

This action has the following format:

```
templategroup action showAttrItem
    name=template_group_name
    [swTemplate=software_template_name]
    jobName=name_of_the_item_order
```

jobName

Specifies the name of the item order to be shown.

name

Specifies the name of the software template, which contains the item order to be shown.

swTemplate

Specifies the name of the software template where the job to be shown is located.

If not coded the default name “<group name> [$date $time]” is taken.

Example: Show Attibutes of Item in software template

Consider to be shown the assigned uninstallation job shown previously. The following command will provide the required information:

```
cadsmcmd templateGroup action=showAttrItem name=tg swTemplate=tg
jobName="prod 1.0/00:unist"


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM
```

```
--------------------------------------------------------------------------------
Show attributes of the assigned job "prod 1.0/00:uninst"
........from the template group "tg"
--------------------------------------------------------------------------------
Job name.............................: prod 1.0/00:uninst
..Item name..........................: prod
..Item version.......................: 1.0/00
..Item type..........................: SOFTWARE
..Procedure name.....................: uninst
..Delivery calendar..................:
..Timeout <days/hours>...............: 7 / 0
..Job startup........................: local time/exact
..Pre action.........................: no action
..Post action........................: no action
..Job operation mask.................: 0
.....................................ignore calendars on target computers
..User parameters....................:
..Prompt timeout <days/hours>.......: 1 / 0
..Job task...........................: 2 – uninstall

SDCMD<A000000>: OK
```

uninstallItem—Uninstall item from template group

This command assigns an *uninstall* job to the software template.

This command has the following format:

```
templategroup action=uninstallItem name=template_group_name
    [swTemplate=software_template_name]
    item=item_name
    version=item_version
    procedure=procedure_name
    [installedWith=install_procedure]
    after={exacttime|boottime}
    [calendarname=delivery_calendar]
    [{preaction={none|reboot|logoff}|bootbefore}]
    [{postaction={none|reboot|logoff|rebootAtEnd|logoffAtEnd|shutdownAtEnd}|boota
    fter}]
    [promptUser[={y|n}]]
```

```
[allowCancel[={y|n}]]
[execTimedOut[={y|n}]]
[prompt=days.hours]
[offline[={y|n}]]
[runAtShutdown[={y|n}]]
[preventLogon[={y|n}]]
[globalTime]
[nocalendar]
[resolveQuery]
[parameters=user_parameters]
[{jobTimeOut=d.h | timeout=hours}]
[userJobMessage]
```

after

Start time of the procedure at the target system.

The following values are valid:

exacttime

The procedure is started immediately after reaching atTime.

boottime

The start of the procedure is delayed until the first boot after atTime has been passed.

allowCancel

If "allowCancel" or "allowCancel=y" is coded, the user will be granted to cancel the job's execution.
If "allowCancel=n" is coded, the user will not be granted.

If the parameter is not coded, the default is given by the related procedure's job option value.

calendarname

Specifies the name of the calendar that controls the time when the evaluation can take place.

execTimedOut

If "execTimedOut " or "execTimedOut=y" is coded, the job's execution will be automatically started when the user prompt times out.
If "execTimedOut=n", the job will not automatically be started.

If the parameter is not coded, the default is given by the related procedure's job option value.

globalTime

If "globalTime" or "globalTime=y" is coded, then the time specifications are based on the enterprise manager time.

If "globalTime=n" is coded, it is based on domain manager time. (default)

installedWith

Specifies the name of the install procedure that has installed the item at the target system.

item

Specifies the name of the item to be uninstalled.

jobTimeout

Specifies the jobTimeout expiration period of a job in the form “d.h” where d specifies the number of days and h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
jobTimeout should not be coded with the timeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 7.0.

name

Specifies the template group name to which the uninstall order is added.

noCalendar

If "noCalendar" is coded, a possible calendar at the target computer will be ignored for this job's execution.

If "noCalendar=n" is coded, the calendar will not be ignored (default).

offline

If "offline" or "offline=y" is coded, the connection between the target computer and the server will be released for the job's execution and reestablished thereafter.
If "offline=n" the connection will not be released.

parameters

Specifies the user parameters for the procedure.
Multiple user parameters must be separated by spaces inside double quotes.
If an empty string is coded, then no parameters are assumed.

If the parameter is not coded, then the default user parameters are given by the procedure definition.

postaction

Specifies the necessary actions that must take place after completion of the specified procedure. The following values are valid:

none

Performs no post-action.

reboot

Restarts the system after completion of the procedure.

logoff

Logs off the user after completion of the procedure.

rebootAtEnd

Restarts the system after completion of all the jobs of this container.

logoffAtEnd

Logs off the user after completion of all the jobs of this container.

shutdownAtEnd

Shuts down the target computer after completion of the jobs.

If the parameter is not coded, the default is given by the related procedure's job option value.

preaction

Specifies that necessary actions must take place before the specified procedure starts. The following values are valid:

none

Performs no pre-action.

reboot

Restarts the system before the start of the procedure.

logoff

Logs off the user before the start of the procedure.

preventLogon

If "preventLogon" or "preventLogon=y" is coded, then the user logon will be rejected during the job's runtime. If a user is already logged on, the execution of the job is delayed until the user logs off.
If "preventLogon=n" the user logon will not be rejected.

If the parameter is not coded, the default is given by the related procedure's job option value.

procedure

Specifies the name of the "uninstall" procedure to be performed.

prompt=d.h

Specifies the time period in which the user will be prompted for the job's start.

The period's format is "*d.h*" where d means days and h means hours.

The range of this parameter is from three hours to seven days (from 0.3 to 7.0). If the specified value falls below the valid minimum, the value is replaced by the minimum. If the specified value exceeds the valid maximum, then the value is replaced by the maximum. No warning is given.

promptUser

If "promptUser" or "promptUser=y" is coded, the user will be prompted for the job's execution.

If "promptUser=n", the user will not be prompted.

If the parameter is not coded, the default is given by the related procedure's job option value.

resolveQuery

Evaluate a possibly related query group before the job is generated.

runAtShutdown

If "runAtShutdown" or "runAtShutdown=y" is coded, the job's execution will start at shutdown only.
If "runAtShutdown=n", the job will start at any other time too.
If the parameter is not coded, the default is given by the related procedure's job option value.
This parameter is for Unix targets only.

swTemplate

Specifies the name of the software template to which the uninstall order is added.

If not coded the default name "<group name> [$date $time]" is taken.

timeout

Specifies the timeout specifies the expiration period of a job in the form “h” where h the number of hours. The range of the parameter is configurable at the manager’s site. When the specified value exceeds the configured upper limit then the value is set to the upper limit, if the specified value goes below the lower limit then the lower limit is taken. No warning is given in any case.
timeout should not be coded with the jobTimeout parameter otherwise an error is reported. If none of the parameters is coded the default is given by 168

version

Specifies the version of the item to be uninstalled.

userJobMessage

Specifies the custom administrator message for an *uninstall* job in a template group.

The parameters "allowCancel," "execTimedOut" and "prompt" are valid, if the "promptUser" option is already set or is set by this command using the "promptUser" parameter.
If neither the "promptUser" option of a job nor the "promptUser" parameter of the command is set, these parameters will be ignored.
No warning is given.

Example:

Again consider the product "prod" of version "1.0/00" and a software template "tg". And at this time an uninstallation of the product should be assigned to the software template. The uninstall routine to be run is named "uninst" and should only run on those systems that have installed the product with the install routine "inst_3". The provided job should meet the following requirements:

- The uninstallation on the targets should be started after the next reboot.
- Calendars assigned to the targets should not be ignored for the uninstallation.
- The user should be prompted for the jobs execution and the user is granted to cancel the job. The job is automatically started after the prompt has timed out. The prompt time out is given 6 hours.

The following command will assign the required job to the specified software template:

```
cadsmcmd templateGroup action=uninstallItem name=tg item=prod version=1.0/00
procedure=uninst installedWith=inst_3 after=boottime promptUser allowCancel
execTimedOut prompt=0.6
```

terminate—Removing Distributions from the Enterprise Database

Terminate is used to remove a failed, halted or completed distribution.

This action has the following format:

`terminate name=`*`distribution_container_name`*

name

Specifies the name of the distribution container to be removed.

uninstall—Generate Uninstall Job in a Job Container

See Install command.

Chapter 4: The command cadsmcmd

The CLI provides support for a number of CA IT Client Manager components like Common Objects (CO), Common Configuration (CCNF), Software Delivery (USD), and OS Installation Management (OSIM).

But the CLI only provides this support if the required services are available at the addressed manager. For example, if SD is not installed at the manager then the CLI will not process any command that requires SD or OSIM services but will report an error (CMD000041).

The CLI offers a number of different interfaces the user might use according to solve his/her automation problems. The syntax of the CLI is as follows:

```
cadsmcmd[local managersystem]
{ batch filename | command | pipe pipename | verbose [loop] }
[login[=loginID[:password]]]
```

The keywords, commands and parameter names of the CLI are case insensitive but "cadsmcmd" itself. Parameter values might be case sensitive.

The cadsmcmd writes its output to stdout (standard output). It can easily be re-directed into a pipe or file for subsequent processing by a script. When cadsmcmd starts it records some general information on stdout that looks as follows:

```
CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved

Trace Mode: Off

Connecting to manager "<default manager>" as user "<default user>" … OK.
Manager: myManager
Domain: myDomain
Domain type: Enterprise
Supporting CO CCNF USD
```

The figures in the second line identify the build number of the cadsmcmd. “Trace Mode” line indicates whether the CLI trace is active or not. In this example the trace is inactive. For more information about the CLI trace see below in chapter “The Trace Option” (see page 517).

The “Connecting” line records how the CLI logs on to the manager. For more details see below in chapters “local” (see page 503) and “login” (see page 516).

The “Manager” records the name of the manager the CLI is connected to, and the “Domain” presents the name of the system the database is located at. This might be the same name if the database is located at the manager.

The “Domain type” line is indicating the type of manager the CLI is connected to. “Enterprise” means that the manager is an CA ITCM enterprise manager while “Domain” indicates a CA ITCM domain manager.

The last line indicates the support provided by the CLI during the current session.

CO means Common Object support and CCNF means Common Configuration support. As CO and CCNF represents base components of CA ITCM they will always be available.

USD is indicating SD support and OSIM is indicating OS Installation Management support.

The behavior of the CLI depends on the options coded when invoked. The meaning of these options as defined above in the syntax definition will be discussed subsequently in more detail.

This section contains the following topics:

local

During the installation the CLI might be assigned to a default CA ITCM manager. When invoking cadsmcmd without the “local” keyword the cadsmcmd connects to this default manager.

If another manager than the default one has to be addressed or if there is no default manager, the “local” keyword has to be used. Separated by at least one blank the “local” is followed by the name of the manager to be addressed (managersystem).

Assume the default manager for a CLI is “defmgr” and another manager is called “mgr_1”. To invoke the verbose interface for the default manager the following has to be coded:

```
cadsmcmd verbose
```

The CLI will record this default manager in his “Connecting” message on stdout.

```
Connecting to manager “<default manager>” as user “<default user>”…
```

To invoke the manager “mgr_1” the following has to be specified:

```
cadsmcmd local mgr_1 verbose
```

The related “Connecting” message looks like

```
Connecting to manager “mgr_1” as user “<default user>”…
```

The manager addressed by the local option can be an enterprise or a domain manager regardless whether the default manager is a domain or an enterprise manager.

Note: The formerly used at-sign “@” for addressing arbitrary managers has become obsolete. If it is coded it is ignored.

batch

Specifying the “batch” keyword the batch interface of the CLI is invoked. In this case the CLI expects a file name (filename) separated by at least one blank from the “batch” keyword. This file name could either be an absolute file name or a relative one. It identifies a file containing a list of CLI commands to be executed by this call.

The CLI processes the batch file as follows. It first establishes a session to the specified manager (for example, a default manager) for the specified user (cf. login (see definition on page 516)). Then it reads the batch file line by line. One line of a batch file can contain more than one command, on the other hand a command can span more than one line. For readability reasons it is recommended to code one command per line. The CLI reads the file from the start until it identifies the next CLI command or it reaches the end of file, what ever comes first. This information read (without the possible command identified at the end) is interpreted as a CLI command with its parameter settings and it is executed.

After a success full execution the scan for the next command or end of file is continued and the scanned command is executed too. This continues until an error occurs or the end of file is reached. The following example presents a contents of a batch file.

```
regsw -item=prod_1 -version=1.0 -path=/products/prod_1 ~procedures=/products/proc_1
regsw -item=prod_2 -version=1.1 -path=/products/prod_2 ~procedures=/products/proc_2
```

This batch file registers two products at the SD. The CLI first scans the file until it finds the second regsw and then processes the regsw command for prod_1 with the parameters item, version, path and procedures. If this succeeds it scans the rest of the file till end of file and processes the second regsw too. If the first regsw fails then the second regsw will not be executed and the CLI terminates.

For documentation purposes it is possible to add comments to a batch file. Comments are indicated by the hash sign “#”. What ever follows a hash sign until the end of a line is understood by the CLI as a comment and is ignored. Due to the special meaning of the “#” it should be avoided in parameter settings. Nevertheless it might happen that a parameter contains a hash sign. In this case enclose the parameter value in quotes to prevent the CLI from interpreting the sign as a comment start. The contents of the file above could therefore turn into the following:

```
# Register product 1
regsw -item=prod_1 -version=1.0 -path=/products/prod_1 ~procedures=/products/proc_1
-comment=”product_1 for department #1”
# Register product 2
regsw -item=prod_2 -version=1.1 -path=/products/prod_2 ~procedures=/products/proc_2
-comment=”product_2 for department #100”
```

By the way not only parameter values containing “#” should be enclosed in quotes but also values containing blanks.

The parameters in the examples above are preceded by a hyphen “-“. It is recommended to indicate parameters in such a way although it is not a must. Nevertheless in case of ambiguous parameters that could be mistaken for a command the hyphen should be set. “stagingServer” is such a parameter that also exists as a CLI command.

As stated above the CLI will normally terminate the batch processing when the execution of a command fails. But sometimes the commands of a batch file are independent of each other. As in the example above the products prod_1 and prod_2 might be independent. So if the registration of prod_1 fails this will have no impact on the registration of prod_2 and therefore the product prod_2 could be registered. By setting the environment variable

```
set SDCMD_CONTINUE=ON (Windows)

export SDCMD_CONTINUE=ON (Linux, bash)
```

the CLI can be forced to ignore failures and to continue processing with the next command from the file. If these variables are set the CLI tries to register prod_2 even if the registration of prod_1 fails.

The CLI records the commands processed and the output provided by the commands on stdout. This output can easily be redirected into a file for further analysis by another script. As an example the

```
002 regsw “item=prod_1” “version=1.0” “path=/products/prod_1”
“procedures=/products/proc_1” “comment=product_1 for department #1”
Command is processed …
SDCMD<A000000>: OK

004 regsw “item=prod_2” “version=1.1” “path=/products/prod_2”
“procedures=/products/proc_2” “comment=product_2 for department #100”
Command is processed …
SDCMD<A000000>: OK
```

The CLI records the commands read and the parameter settings as it is understood by the CLI. The figure in front of the command refers to the line in the batch file where the command has been detected.

All valid commands of the CLI are available at the batch interface.

pipe

The pipe interface offers the advantage of a batch processing and a direct command invocation, i.e., it is possible to run more than one command during a session between the CLI and the manager as with the batch option, but on the other hand a script has the chance to process the command output before it launches the next CLI command. The pipe interface is activated by the pipe keyword followed by the name of the pipe the CLI has to listen to. The pipe specified is a named pipe on Windows while it is a fifo file on Linux. In any case, the CLI works as a pipe client, i.e., the pipe has to be created and administered by the invoking script. When invoked for pipe interface processing, the CLI establishes the connection to the manager specified with the credentials specified and records the result to stdout. When the session is established, the CLI listens to the pipe for any command to be processed. When it receives one the command is processed, and its results are recorded at stdout. After the command is processed, the CLI listens again for the next command. Differing from the batch mode, the CLI in pipe mode will always return to the pipe regardless whether a command has been processed successfully or not; and the session between CLI and manager will not be terminated when a command fails. This behaviour corresponds to the batch mode with SDCMD_CONTINUE=ON.

To ease the communication and synchronization between the CLI and the launching script, the CLI provides certain eye catchers that indicate the end of a CLI processing and that the CLI now starts listening to the pipe again.

When starting the CLI in pipe mode, the CLI launches the message on stdout

```
*** SDCMD: Line up ***
```

when the session to the manager is established and the CLI starts listening to the pipe. From this time on a script can start sending commands via pipe to the CLI. Also, the end of a command processing is reported by a special eye catcher. Whenever a command terminates, successfully or not, the CLI launches the following message on stdout:

```
### SDCMD: -eoc ###
```

When a script detects this string, then it knows that the command launched by it has terminated and all command output is written to stdout.

The commands to be sent by an application via pipe are of the same format as in batch files. The command buffer written to the pipe by the application must contain the complete command to be executed. A command can not be distributed among several send buffers.

On the other hand, it is possible to send more than one command within one buffer. In this case the commands have to be separated by the following string:

```
-eoc
```

The session between a CLI in pipe mode and the manager is terminated as soon as the CLI receives the string

```
-quit
```

via pipe.

The CLI is invoked for the pipe interface by coding the “pipe” keyword followed by the name of the pipe (pipename) the CLI uses to receive its commands to be processed. Between the keyword and the pipename has to be at least one blank.

The following example shows the usage of the pipe interface on Linux. The script lists all targetComputers registered at the USD and all products registered at the USD software library.

```
#! /usr/bin/sh
proto ()
{
        aa=" "
        bb=`echo "$aa" | grep "SDCMD: Line up"`
        while [ -z "$bb" ]
        do
                read aa
                bb=`echo "$aa" | grep "SDCMD: Line up"`
                if [ -z "$bb" ]
                then
                        echo "$aa"
                fi
        done
}
prot ()
{
        aa=" "
        bb=`echo "$aa" | grep "SDCMD: -eoc"`
        while [ -z "$bb" ]
        do
                read aa
                bb=`echo "$aa" | grep "SDCMD: -eoc"`
                if [ -z "$bb" ]
                then
                        echo "$aa"
                fi
        done
}

mkfifo /ca_dsmcmd_in
mkfifo /ca_dsmcmd_out
```

```
{
        cadsmcmd pipe /ca_dsmcmd_in > /ca_dsmcmd_out &
        proto
        echo "targetComputer action=list -eoc" > /ca_dsmcmd_in
        prot
        #
        #  store returned list and process it
        #
        #     ...
        #
        #  next command
        echo "swLibrary action=list -eoc" > /ca_dsmcmd_in
        prot
        #
        #  store returned list and process it
        #
        #      ...
        #
        #  terminate session
        echo "-quit" > /ca_dsmcmd_in
} < /ca_dsmcmd_out
rm /ca_dsmcmd_*
```

The example consists of two routines and the mainline part. The routine “proto ()” triggers for the “### SDCMD: Line up ###” indicating the pipe interface being up and waiting for commands and the routine “prot ()” triggers for “### SDCMD: -eoc ###” indicating processing complete for a command.

The mainline part first creates two pipes. The pipe “/ca_dsmcmd_in” is used by the CLI as input pipe, i.e., the CLI is receiving the commands to be processed via this pipe. The other pipe “/ca_dsmcmd_out” is used to redirect the stdout of the CLI to the script so that the script might be able to process the results. After these pipes have been created a block is started that starts the CLI in pipe mode in batch, launches two list commands for targetComputer and swLibrary, and after that terminates the session. This block receives its input from the “/ca_dsmcmd_out” pipe recording the output of the CLI. It is strongly recommended to keep this pipe open for the script during the life time of the CLI. That is why it is assigned as input for this block. If, for example, this pipe is only assigned as input to the proto and prot procedures, the user might experience run time problems that result in loss of data and might be in hanging sessions.

After the CLI is launched in pipe mode in batch the proto routine is used to trigger for the CLI becoming ready for action. When it returns the script launches the list action for targetComputer and then uses the prot routine to determine the end of command processing. The output of this prot routine could be redirected into a file for subsequent script processing. Then the list action for swLibrary is launched and again the prot routine is used to determine the end of command processing. And again the output of the prot procedure could be redirected into a file for subsequent script processing. Finally the block is terminated by shutting down the CLI per -quit.

The last line of the script is for housekeeping and cleaning up the pipes.

The following example does the same as the example above, but it uses DM Scripting and is for Windows.

```
Dim hPipeIn, hPipeOut As Integer
Dim pipeIn, pipeOut As String
Dim command_1, command_2, eoc, eop, lineUp, endOfCommand, buffer As String

' ********************************************************************
' clear screen
ClrScr()

' ********************************************************************
' set constants
eoc = " -eoc"
eop = "-quit"
command_1 = "targetComputer action=list" + eoc
command_2 = "swLibrary action=list" + eoc
lineUp = "SDCMD: Line up"
endOfCommand = "SDCMD: -eoc"
```

```
pipeIn ="\\.\pipe\ca_dsmcmd_in"
pipeOut ="\\.\pipe\ca_dsmcmd_out"

' ************************************************************************
' Create input pipe of CLI
hPipeIn = CreatePipe(pipeIn, O_WRITE)
If (hPipeIn = -1) Then
        MsgBox("Open pipe """ + pipeIn + """ failed.", "Pipe test", MB_OK +
                                MB_ICONSTOP + MB_SETFOREGROUND)
        exit
EndIf

' ************************************************************************
' Create output pipe of CLI
hPipeOut = CreatePipe(pipeOut, O_READ)
If (hPipeOut = -1) Then
        MsgBox("Open pipe """ + pipeOut + """ failed.", "Pipe test", MB_OK +
MB_ICONSTOP + MB_SETFOREGROUND)
        CloseFile(hPipeIn)
        exit
EndIf

' ************************************************************************
' Launch cadsmcmd in pipe mode using pipeIn as input pipe and redirecting
‘ stdout to pipeOut
Execute("cmd /c ""cadsmcmd pipe " + pipeIn + " > " + pipeOut + """",
                                FALSE, 0)
While (TRUE)
        If (Not(ReadFile(hPipeOut, buffer))) Then
                MsgBox("Read from pipe """ + pipeOut + """ failed.", "Pipe test",
                                        MB_OK + MB_ICONSTOP + MB_SETFOREGROUND)
                CloseFile(hPipeIn)
                CloseFile(hPipeOut)
                exit
        EndIf
        If (InStr(buffer, lineUp) = 0) Then
                Print(buffer)
        Else
                ExitWhile
        EndIf
Wend
```

```
' ************************************************************************
' launch request for a targetComputer action=list
If (Not(WriteFile(hPipeIn, command_1))) Then
        MsgBox("Write to pipe """ + pipeIn + """ failed.", "Pipe test", MB_OK +
                        MB_ICONSTOP + MB_SETFOREGROUND)
        CloseFile(hPipeIn)
        CloseFile(hPipeOut)
        exit
EndIf
While (TRUE)
        If (Not(ReadFile(hPipeOut, buffer))) Then
                MsgBox("Read from pipe """ + pipeOut + """ failed.", "Pipe test",
                                        MB_OK + MB_ICONSTOP + MB_SETFOREGROUND)
                CloseFile(hPipeIn)
                CloseFile(hPipeOut)
                exit
        EndIf
        If (InStr(buffer, endOfCommand) = 0) Then
                ' results could be stored for subsequent processing or directly
processed.
                Print(buffer)
        Else
                ExitWhile
        EndIf
Wend

' ************************************************************************
' launch request for a swLibrary action=list
If (Not(WriteFile(hPipeIn, command_2))) Then
        MsgBox("Write to pipe """ + pipeIn + """ failed.", "Pipe test", MB_OK +
                        MB_ICONSTOP + MB_SETFOREGROUND)
        CloseFile(hPipeIn)
        CloseFile(hPipeOut)
        exit
EndIf
While (TRUE)
        If (Not(ReadFile(hPipeOut, buffer))) Then
                MsgBox("Read from pipe """ + pipeOut + """ failed.", "Pipe test",
                                        MB_OK + MB_ICONSTOP + MB_SETFOREGROUND)
                CloseFile(hPipeIn)
                CloseFile(hPipeOut)
                exit
```

```
        EndIf
        If (InStr(buffer, endOfCommand) = 0) Then
                ' results could be stored for subsequent processing or directly
processed.
                Print(buffer)
        Else
                ExitWhile
        EndIf
Wend

' ************************************************************************
' terminate CLI
If (Not(WriteFile(hPipeIn, eop))) Then
        MsgBox("Write to pipe """ + pipeIn + """ failed.", "Pipe test", MB_OK +
                        MB_ICONSTOP + MB_SETFOREGROUND)
        CloseFile(hPipeIn)
        CloseFile(hPipeOut)
        exit
EndIf

' ************************************************************************
' clean up
CloseFile(hPipeIn)
CloseFile(hPipeOut)
```

All valid commands of the CLI are available at the pipe interface.

command

The cadsmcmd can be directly invoked with a CLI command and its parameter settings required. The CLI establishes a session to the addressed manager, processes the command, and terminates the session. The CLI records the command output and status information on stdout. Launching the following command provides the subsequent output:

```
cadsmcmd targetComputer action=list



CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off
```

```
Connecting to manager "<default manager>" as user "<default user>" ... ok.
Manager: mymanager
Domain: mydomain
Domain type: Domain
Supporting: CO CCNF USD OSIM


------------------------------------------------------------------------
List of target computers
------------------------------------------------------------------------
mgr_1(scalability server; Win Server 2003 Intel; Boot server)
mgr_1/Administrator(computer user; Win Server 2003 Intel)
scal_2(scalability server; Win 2000 Intel; Boot server)
scal_2/Administrator(computer user; Win 2000 Intel)
comp_0(computer; Any)
comp_1(computer; Win XP Intel)

------------------------------------------------------------------------
List of un-named and OSIM un-managed computers
------------------------------------------------------------------------
FF:FF:42:65:13:7E(---; ---; BMS un-managed, mgr_1)
FF:FF:42:65:13:83(---; ---; BMS un-managed, scal_2)

------------------------------------------------------------------------
Total number of target computers: 6
Number of un-named and OSIM un-managed systems shown: 2
SDCMD<A000000>: OK
```

The commands and their valid parameters are described subsequently.

verbose

The verbose interface offers the user the possibility to create and execute CLI commands in a dialog manner. The interface is invoked by specifying the keyword “verbose”. The CLI establishes a session to the addressed manager and determines the type of manager, i.e. whether it is a CA ITCM Domain Manager or a CA ITCM Enterprise Manager. According to this type the CLI offers the user a list of commands that are valid for this type of manager. For a domain manager running SD the offering looks like

```
Valid Commands are:
=======================
activate........aregsw
cancel..........comconf
compgroup.......configure
dereg...........deregproc
install.........jobContainer
```

```
managedcomputer recover
regproc.........regsw
renewjob........setdbcredentials
stagingserver...swLibrary
targetComputer..templateGroup
uninstall

quit
======================
Please, enter one of the commands above.............:
```

The user can select one of the offered commands and the CLI will guide the user through the valid parameters and their settings. Only those parameters are presented that are needed due to the already entered information. When all the needed information is acquired the CLI will execute the command. The output of the command as well as status information is written on stdout.

When the “loop” option is not coded, the CLI will terminate after the specified command has been processed. If it is coded, it returns to the command selection list, so that the user can specify and run the next command. Entering the “quit” command will terminate such a session. This quit command is valid in the verbose mode only.

All valid commands of the CLI are available at the verbose interface.

loop

(Optional) Normally the verbose mode allows selecting one command per cadsmcmd invocation. Coding "loop" will allow more than one command.
The loop command is valid in the verbose mode only.

quit

When using "loop", entering the “quit” command will terminate the CLI session. The quit command is valid in the verbose mode only.

Non Privileged Users

On Windows the cadsmcmd can be used by non privileged users without any adaptations.

On Linux a group “sdadmins” has to be provided and only those users assigned to this group are allowed to run cadsmcmd, other users will receive the message “Access denied” when launching cadsmcmd.

login

The “login” keyword enables the user to log on to the manager and run the CLI commands as a different user than the default user. The default user is representing the user as he/she is currently logged on at the console from where the CLI command is launched. For example, being logged on to system as user “hugo” and launching the following command:

```
cadsmcmd local mgr_1 verbose
```

The CLI connects to the manager “mgr_1” for the “default user”. The CLI records this in the “Connecting” message:

```
Connecting to manager “mgr_1” as user “<default user>”…
```

It tries to establish the session to the manager as the current user “hugo”. If this user is not known to mgr_1 or has a different password than the session establishment will fail. In this case different credentials have to be used to authenticate the session.

```
login[=loginID[:password]]
```

The information passed with the “login” keyword has changed with CA ITCM Release 12.8 according to a different authentication method used by CA ITCM Release 12.8 compared to USD 4.0. The login process has to be supplied with information about the security provider, the security authority, the user id, and the user password. The password is passed with the password parameter while the security provider, the security authority and the user id are passed with loginID. The loginID is build as

```
\”<security provider>://<security authority>/<user id>\”
```

Note: The quotes (”) are mandatory. Otherwise the CLI is not able to process the login information correctly. As the quotes has to be interpreted by the cadsmcmd and not by the shell or command prompt they should be protected by a preceding backslash (\).

A security provider could be a Windows system identified by “winnt” or a Linux system identified by “unixl”. The security authority could be a domain or the addressed system. Consider the CLI should login with the verbose interface to Linux system “lx_sy_1.myCo.com” as “root” with password “5k7zpq]r”. The following command should be launched:

```
cadsmcmd local lx_sy_1 verbose login=\”unixl://lx_sy_1.myCo.com/root\”:5k7zpq]r
```

The CLI will record this by the message:

```
Connecting to manager “lx_sy_1” as user “unixl://lx_sy_1.myCo.com/root”…
```

In case that the verbose interface should be started for the Windows manager “win_sy_01” and user “myid” for the password “7hj.98,a” the launched command might look as follows:

```
cadsmcmd local win_sy_01 verbose login=\”winnt://win_sy_01/myid\”:7hj.98,a
```

In this case the CLI will record

```
Connecting to manager “win_sy_01” as user “winnt://win_sy_01/myid”…
```

If login is coded with loginID but no password then the CLI assumes that there is no password for the specified user. If login is coded without userID and password then the CLI will start a short dialog for acquiring these parameters. In this case the password is not shown on the console when entered.

Note: When cadsmcmd is invoked with the login parameter the local parameter should be coded too to enhance security.

The trace option

The CLI offers a trace for debugging the CLI’s internal workflow in case of problems. By default the trace is off. It should only be activated on request of the CA support. The trace is driven by the environment variables SDCMD_TRACE and SDCMD_FILE. On Windows the syntax for setting SDCMD_TRACE is as follows

```
set SDCMD_TRACE={all | file | screen}
```

and on Linux for the bash:

```
export SDCMD_TRACE={all | file | screen}
```

If “screen” is coded, the CLI will record its trace entries on stdout i.e. the console from where it has been started.

If “file” is coded for the environment variable, then the CLI records into a file.

This file is specified by SDCMD_FILE variable and can be specified as follows:

```
set SDCMD_FILE=<file name>
```

on Windows

and

```
export SDCMD_FILE=<file name>
```

on Linux bash

If SDCMD_FILE is not set then the CLI writes into the file cadsmcmd.log of the CA ITCM log directory.

In case of “all” the trace is written into the specified file and on stdout.

To turn off an activated trace, enter the following commands:

```
set SDCMD_TRACE=
set SDCMD_FILE=
```

for Windows and

```
unset SDCMD_TRACE
unset SDCMD_FILE
```

for Linux bash.

Whether a trace is activated for the CLI is recorded at the CLI start.

```
c:\>cadsmcmd verbose


CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: FILE
Trace file name: c:\temp\tst.txt

Connecting to manager "<default manager>" as user "<default user>" ...OK.
```

In case of “Trace mode” SCREEN the “Trace file name” line is omitted. In case of no active trace the following is recorded:

```
c:\>cadsmcmd verbose

CA IT Client Manager r12
ITCM Command Line Version 12.8.0.xxxx
Copyright (c) 2013 CA. All rights reserved.

Trace mode: Off

Connecting to manager "<default manager>" as user "<default user>" ...OK.
```

Fonts

When the output of a CLI commands contains some local characters then the shell processing this command should be enabled to present these characters correctly, for example, supporting German characters on Windows the font should be changed to Lucida Console.

Parameters

All parameter names of the commands are case-insensitive, however the values assigned to them may not be. For example, a path name on Windows might be case-insensitive, on Linux it will definitely not be case-insensitive.

As mentioned above in the chapter dealing with the batch interface, parameters can be preceded by a hyphen (-) but this is optional. When coding a batch file it is recommended to use this preceding hyphen to avoid any ambiguity.

The CLI uses the blank as a delimiter to separate parameters, options and commands. If a parameter value contains a blank, then the parameter value should be enclosed in quotes. As these quotes should not be interpreted by a shell or command prompt but by the CLI, it might be necessary to escape the quotes from a shell or command prompt by preceding backslashes (\).

Note: The CLI reports errors only on parameters not known to the CLI or to inconsistent parameters. If a parameter is coded with a command or action that is not mentioned in the syntax description (but it is a defined parameter of the CLI and used by other commands), then the parameter is ignored for the coded command or action. No error or warning is displayed.

Length Restrictions

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

Parameter	Maximum Length
Distribution Name	129
Error Message	512
Software Name	129
Software Version	129
Item Procedure Name	129
Area or Domain Name	129
Computer Name	129
Computer Group Name	129
Supplier Text	255
Comment	255
Source Path	255
Query Name	255
Computer Group Comment	255
Domain Comment	255

Restrictions

To avoid any ambiguities with the CLI the following names or labels used with the following objects should be unique:

- targetComputers
- computerGroups
- software policies or templates
- serverGroups

Using Filters

Filters are used by some list methods to confine the amount of information listed.

The filters are expressions constructed from atomic expressions using operators of propositional logic like && (and), || (or), and ! (not), following the subsequent rules:

- Each atomic expression is a filter expression
- If exp is a filter expression then !exp is a filter expression.
- If exp1 and exp2 are filter expressions then (exp1 && exp2) and (exp1 || exp2) are filter expressions too.

The valid atomic expressions depend on the objects to be listed. These atomic expressions have the following Syntax

```
attributename operand value
```

The valid attribute names for new and already existing list methods are listed in the chapters following.

There are different types of attributes:

Type	Meaning
String	The attribute value is of type string
Long	The attribute value is a 4 byte integer
Date	The attribute value is of date type YYYY-MM-DD where YYYY indicates the year, MM the month and DD the day of month
Time	The attribute value is of time type hh:mm where hh indicates the hour in 24 hour notation and mm the minutes
DateTime	Is a combined date and time notation YYYY-MM-DD hh:mm according to ISO standard.

The following operands are valid and are true:

Operand	String	Long
=	If the attribute value matches the specified value	If the attribute value is equal to the specified value
!=	If the attribute value does not match the specified value	If the attribute value is not equal to the specified value

Operand	String	Long
<	If the attribute value is lexicographically less than the specified value	If the attribute value is less than the specified value
>	If the attribute value is lexicographically greater than the specified value	If the attribute value is greater than the specified value

In case of "=" or "!=" the specified value may contain the wild character '*' to indicate an arbitrary string, even an empty one.

Attributes of type Date, Time or DateTime are handled like attributes of type string.

Sometimes attributes allow a union of distinct values only. Unless stated otherwise, these attributes are handled like attributes of type string.

Usage:

```
cadsmcmd compgroup action=list filter="(Group type=Group && Scope=local)"
```

In case the filter is created using a single expression, you need not use the brackets.

Note: If an attribute in an atomic expression is misspelled or an attribute does not apply to an object, the atomic expression evaluates to true unless stated otherwise.

Chapter 5: dmsweep Commands

dmsweep commands are valid on Windows (except Windows NT 4.0, Windows 98, Windows 95), and Linux platforms.

dmsweep is the command line interface to the infrastructure deployment components of DSM. You can use dmsweep to initiate deployment of DSM agent and server packages to target computers, and scan for the presence of DSM software on target computers. You can use dmsweep to manage DSM agent and server packages on scalability servers.

The dmsweep command line has the following format:

```
dmsweep <subcommand> [{</option> [<arg>]}*]
```

subcommand

One of the following subcommands:

deploy (see page 524)
help (or usage) (see page 524)
packages (see page 530)
scan (see page 533)
sspack (see page 536)
stage (see page 539)
version (see page 542)

For details of valid values for subcommand and option, see dmsweep Subcommands (see page 524).

option

An option for the subcommand. All options are preceded by a slash (/) or hyphen (-).

Most options have a long and short form that you can use interchangeably, for example, you can use /tu instead of /tusername. For more information, see Abbreviations for dmsweep Arguments (see page 545).

arg

An argument for the option.

Options that take an argument take the following formats:

```
</option> <arg>  (use a space to separate arguments)
</option>=<arg>
</option>:<arg>
```

If an argument contains spaces you must enclose it in quotation marks; for example

```
/option="my arg"
```

If an option allows multiple arguments they must be comma-separated (with no surrounding spaces), for example:

```
dmsweep scan /ip=10.0.0.1,10.0.0.2,10.0.0.3
```

dmsweep Subcommands

The following sections describe the dmsweep subcommands. The description of each subcommand includes details of valid options for the subcommand.

dmsweep help, dmsweep usage

Displays a message describing all the dmsweep subcommands and their syntax. This message is also output if the dmsweep command is executed without arguments.

The subcommand has the following format:

```
help
usage
```

Example:

The following example shows how you can use the dmsweep help command line utility to display subcommands, options, and arguments.

To display the dmsweep command options enter any of the following:

```
dmsweep
dmsweep help
dmsweep usage
```

dmsweep deploy

The dmsweep deploy command attempts deployment of the specified package to the specified list of target computers.

The subcommand has the following format:

```
deploy
    [/manager <manager_machine>]
    [/musername <manager_username> /mpassword <manager_password>]
    [/mfilename <manager_filename>]
    {/ip <address_wild> [/toip <stop_address>] | /ipfilename <ip_filename> | /domain
    <domain_name> (WinNT only)
    | /uri <uniform resource identifier>
    | /targetcred <target_credentials_file>
```

```
| /query <query_name> }
[/tusername <target_username> /tpassword <target_password> | /epassword
<encrypted_password>]
[/tfilename <target_filename>] |
{/packagenum <package_number> | /product <product_name> /package <package_name>
[/version <version>] [/platform <platform>] [/language <language>]}
[/pparams <package_parameters>]
[/jobname <job_name>]
[/sserver <scalability_server_machine>
 /ssusername <scalability_server_username> /sspassword
<scalability_server_password>]
 [/primerargs <primer install arguments>]
```

/manager

Specifies the name of the manager from which to retrieve the package list. If you omit /manager, the current machine is assumed.

/musername

Specifies the username for the manager computer.

/mpassword

Specifies the password for the manager computer.

If <manager_password> is not supplied, dmsweep will prompt the value when the command is executed.

/mfilename

Lets you read the name of the manager and details of its username and password from a file.

The format of this file is described in the packages section.

/ip

Start address from which to scan.

You can specify an address as a machine name, if it can be successfully resolved (for example, by DNS), or as an IP address (for example 10.0.0.1). The <address_wild> argument enables you to use asterisks (for example, 10.0.0.* covers the range 10.0.0.1 to 10.0.0.254 and 10.0.*.* covers the range 10.0.0.1 to 10.0.255.254.

You can specify a single address for <address_wild> or a comma-separated list of addresses.

/toip

End address for the range to scan.

If you specify /toip in conjunction with /ip, the scan starts with the address specified by <address_wild> and stops after scanning <stop_address>.

/ipfilename

If you specify /ipfilename, the command reads the list of target machines is from the file <ip_filename>. Each line of the file contains a single target address or multiple addresses separated by commas, for example:

machine1

machine2, machine3, 10.0.0.1

machine5

/domain

If a deployment manager is on a Windows-based computer, you can use /domain to scan the computers in the specified domain.

/uri

If you specify /uri you can use a uniform resource identifier to specify target computers to read from an external directory.

/targetcred

If you specify /targetcred, the command reads the list of target machines and connection credentials from the file <target_credentials_file>.

/query

If /query is specified, the list of targets will be generated when the unmanaged asset query identified by <query_name> is run.

/packagenum

Package number of the package to be scanned for. The output from the dmsweep packages command describes the valid values.

Note: Use of the /packagenum option allows rapid specification of packages, but be aware that the package numbers may differ as new packages are added to the deployment package library. If you wish to identify packages in a non-variant way (for example when writing batch scripts), use the package identification options below.

/product

The product name (for example "CA Unicenter DSM") to be scanned for. The output from the dmsweep packages command describes the valid values.

/package

The package name (for example "Agent + all agent plugins") to be scanned for. The output from the dmsweep packages command describes the valid values.

/version

Version of the package (for example "11.2.0.1234") to be scanned for. The output from the dmsweep packages command describes the valid values. Optional but if you do not supply enough information to uniquely identify the package an error is displayed.

/platform

Platform property of the package (for example "Linux x86") to be scanned for. The output from the dmsweep packages command describes the valid values. Optional but if you do not supply enough information to uniquely identify the package an error is displayed.

/language

Language property of the package (for example "ENU") to be scanned for. The output from the dmsweep packages command describes the valid values. Optional but if you do not supply enough information to uniquely identify the package an error is displayed.

/tusername

Specifies the username for connecting to the target. If your deployment manager has already deployed packages to the target computer, credentials may not be required.

/tpassword

Specifies the password for connecting to the target. If your deployment manager has already deployed packages to the target computer, credentials may not be required.

If you specify /tusername but omit <target_username>, the user is prompted for the username. If you specify /tpassword but omit <target_password>, the user is prompted for the password, which will not be displayed on the screen.

/epassword

Specifies the password in encrypted form. This option alerts dmsweep to this format and the password will be decrypted before being used. See Encrypted Passwords (see page 542).

/tfilename

If a target needs authentication and /tfilename has been specified, <target_filename> is searched. If an entry exists, the credentials specified in the file are used for the target, otherwise /tusername and /tpassword are used. The file <target_filename> contains one target per line, with each line containing the target's machine name (or IP address), username and password, for example:

10.0.0.1 username1 password1

10.0.0.2 username2 password2

* username3 password3

An entry with '*' as the target name denotes default credentials for any target not explicitly listed in the file.

/pparamsT

Specifies a comma-separated list of parameters to replace 1, 2 and so on on the package install command line. To display the command line along with the expected format and description of each parameter use dmsweep's packages command. For a list of installation options valid for the different agent and server deployment packages, see the CA ITCM Implementation Guide.

/primerargs

Specifies non-default installation arguments (for example non-default installation locations) to the dmprimer installation (but not the agent or server installation). The install arguments are passed unmodified to the installation command line on the computer on which a primer is to be installed. For a list of valid installation options, see the CA ITCM Implementation Guide.

/jobname

Specifies the name of the job is set to the supplied <job_name> argument; otherwise it will be generated automatically based on the current date and time.

/sserver

Forces the target machines to obtain their packages from a scalability server. If not specified the package is obtained from the manager machine.

/ssuserame

Specifies the username for connecting to the scalability server.

/sspassword

Specifies the password for connecting to the scalability server.

Note: If you specify /ssusername but omit <scalability_server_username>, the user is prompted for the username. If you specify /sspassword but omit <scalability_server_password>, the user is prompted for the password, which will not be displayed on the screen.

Note: You can use <target_credentials_file> instead of <ip_filename> and <target_filename>. For details of /targetcred, see The Target Credentials File.

Examples:

Following you will find some examples showing the various possibilities of the deploy subcommand:

- To deploy to a target through a local manager use the command

  ```
  dmsweep deploy /ip <target machine > /pn 2 /pparams <myserver>,,
  ```

- To deploy a package to a target computer that needs credentials using the local manager, enter the following command:

  ```
  dmsweep deploy /ip <machine name> /pn 2 /pparams <myserver>,, /tu <target username> /tp <target password>
  ```

- To deploy a package to a target computer that needs credentials wherein the password may be encrypted, enter the following command:

  ```
  dmsweep deploy /ip <machine name> /pn 2 /pparams <myserver>,, /tu <target username> /ep <encrypted password>
  ```

- The following example shows how you can use the dmsweep command line utility to prompt for username and password.

  dmsweep has a password prompting feature, which enables you to enter passwords without the password appearing on the screen. You can also prompt for usernames but they are displayed as they are typed. To prompt for the username or password, enter the option but do not supply a value, as shown in the following example.

  To deploy a package to a target using the local manager with password and username prompting, enter the following command:

  ```
  dmsweep deploy /ip <target machine> /pn 2 /pparams <myserver>,, /tu  /tp
  ```

- To deploy a package to a range of machines, where no credentials are required, enter the following command:

  ```
  dmsweep deploy /ip 192.0.25.1 /toip 192.0.25.9 /pn 3 /pparams <myserver>,,
  ```

- To deploy to all the machines in a subnet (192.0.45) using a wildcard, enter the following command:

  ```
  dmsweep deploy /ip 192.0.45.* /pn 3 /pparams <myserver>,,
  ```

  You can also use wildcards to deploy to multiple subnets, enter the following command:

  ```
  dmsweep deploy /ip 192.0.*.* /pn 1 /pparams <myserver>,,
  ```

- To deploy to a target through a remote manager, enter the following command:

  ```
  dmsweep deploy /ip <target machine> /pn 3 /mgr <remote manager> /mu <remote manager user> /mp <remote manager password> /tu <target user> /tp <target password>
  ```

- To deploy to a target through a remote manager but with username and password prompting for both the manager and target machines, enter the following command:

  ```
  dmsweep deploy /ip <target machine> /pn 3 /mgr <remote manager> /mu  /mp  /tu /tp
  ```

Note: Replace values such as the package number, target computers and manager machines with suitable values.

dmsweep packages

The dmsweep packages subcommand displays the list of packages available for deployment on the specified deployment manager.

"packages" can be shortened to "pack", that is, you can use the command "dmsweep pack".

The subcommand has the following format:

```
packages
    [/manager <manager_machine>]
    [/musername <manager_username> /mpassword <manager_password>]
    [/mfilename <manager_filename>]
```

/manager

Specifies the name of the manager from which to retrieve the package list. If you omit /manager, the current machine is assumed.

/musername

Specifies the username for the manager computer.

The user will be prompted for the username if /musername is entered without <manager_username>.

/mpassword

Specifies the password for the manager computer.

If <manager_password> is not supplied, dmsweep will prompt the value.

Note: Neither /musername nor /mpassword can be specified without /manager.

/mfilename

Lets you read the name of the manager and details of its username and password from a file.

If the manager requires authentication and /mfilename has been specified, the <manager_filename> is searched. If an entry exists for the manager, the credentials in the file are used, otherwise <manager_username> and <manager_password> are used. The file <manager_filename> lists one manager per line with each line containing the manager's machine name, username and password; for example:

```
manager1 username1 password1
manager2 username2 password2
* username3 password3
```

An entry with '*' as the manager name denotes default credentials, which are used for any manager name not explicitly listed in the file.

If the /mpassword option is specified but <manager_password> is omitted, the password is prompted for when the command runs.

Example:

The following example shows how you can use the dmsweep command line utility. Replace values such as the package number, target computers and manager machines with suitable values.

To list the packages available for deployment on the local machine, enter one of the following:

```
dmsweep packages
dmsweep pack
```

Output from the command appears. The following extract is part of the example output, and it shows the package numbers that you can use with the /packagenumber (or /pn) options in subsequent commands.

```
Contacting the deployment manager on ASTEROID.
New job id is 13191950.
Retrieving the package list.
Found 5 packages on deployment manager ASTEROID:

..Package 1:
....Product.....: CA DMS
....Package.....: Agent + Asset Management Plugin
....Version.....: 11.1.8124.2426
....Platform....: Windows_x86
    Language    : ENU
....Description.: Deploy agent with Asset Management plugin.
....Command.....: AgtAM.msi"  AGENT_SERVER=$1$ ALLUSERS=1 ENSURE_CAF_STOPPED=1
CAF_COMMON_CONFIG=1 CAF_INSTALL_SERVICE=1 CAF_START_SERVICE=1 /qn /l*v
%TEMP%\ITRMSetupAgtAM.log EXITFILE=$E$ $2$
....UNIX Command : installdsm" -r $P$/install.rsp /REXITFILE=$E$ /RITRM_SERVER=$1$
$3$

....Parameter 1 : Please enter the Scalability Server address to connect to

....Regular Expression Validation : ^[^<>`~!/@\#}$%:;)(_^{*=|'+]{1,255}$
```

```
....Parameter 2 : Please enter any additional Windows install options
....Regular Expression Validation : .*

..Package 2:
........
........:
```

In the example output, there are two parameters listed for package 1. If you select this package for deployment you must supply two parameters using the /pparams option. Use commas to separate the parameter values.

For example, to select package 1 for deployment, you could use the command

```
dmsweep deploy /pn 1 /ip 192.0.2.5.9 /pparams <myserver>,"CA=D:\DSM"
```

where parameter 1 (the Scalability Server address) is set to <myserver>, parameter 2 (the Windows install options) has the value "CA=D:\DSM", which requests the package should be installed in a non-default location.

The parameters appear in the DSM Explorer in three separate fields on the Agent Configuration page when you use the DSM Explorer to deploy a package.

The Regular Expression Validation string specifies the range of characters that are valid for the associated parameter. A validation string of '.*' means there are no syntactic restrictions on the parameter value.

dmsweep scan

The dmsweep scan command scans the specified target addresses, determines which addresses are active and available for deployment, and determines if the target computers have the requested package installed.

The subcommand has the following format:

```
scan [/manager <manager_machine>]
    [/musername <manager_username> /mpassword <manager_password>]
    [/mfilename <manager_filename>]
    {/ip <address_wild> [/toip <stop_address>]
```

```
 | /ipfilename <ip_filename>
 | /domain <domain_name> (WinNT only)
 | /uri <uniform resource identifier>
 | /targetcred <target_credentials_file>
 | /query <query_name>}
{/packagenum <package_number>
 |  /product <product_name> /package <package_name>
 [/version <version>]
 [/platform <platform>]
 [/language <language>]}
```

/manager

Specifies the name of the manager from which to retrieve the package list. If you omit /manager, the current machine is assumed.

/musername

Specifies the username for the manager computer.

If <manager_username> is omitted the user will be prompted for it.

/mpassword

Specifies the password for the manager computer.

If <manager_password> is not supplied, dmsweep will prompt the value.

/mfilename

Lets you read the name of the manager and details of its username and password from a file.

/ip

Start address or addresses from which to scan.

You can specify an address as a machine name, if it can be successfully resolved (for example, by DNS), or as an IP address (for example 10.0.0.1). The <address_wild> argument enables you to use asterisks (for example, 10.0.0.* covers the range 10.0.0.1 to 10.0.0.254 and 10.0.*.* covers the range 10.0.0.1 to 10.0.255.254.

You can specify a single address for <address_wild> or a comma-separated list of addresses.

Note: This parameter is the start address only when the /toip option is also used.

/toip

End address for the range to scan.

This option can only be used in conjunction with /ip. When used the scan starts with the address specified by <address_wild> and stops after scanning <stop_address>.

/ipfilename

If you specify /ipfilename, the command reads the list of target machines is from the file <ip_filename>. Each line of the file contains a single target address or multiple addresses separated by commas, for example:

machine1

machine2, machine3, 10.0.0.1

machine5

/domain

If a deployment manager is on a Windows-based computer, you can use /domain to scan the computers in the specified domain.

/uri

If you specify /uri you can use a uniform resource identifier to specify target computers to read from an external directory.

/targetcred

If you specify /targetcred, the command reads the list of target machines and connection credentials from the file <target_credentials_file>.

/query

If /query is specified, the list of targets will be generated when the unmanaged asset query identified by <query_name> is run.

/packagenum

Package number of the package to be scanned for. The output from the 'dmsweep packages' command describes the valid values.

/product

Property of the package to be scanned for. The output from the dmsweep packages command describes the valid values.

/package

Property of the package to be scanned for. The output from the dmsweep packages command describes the valid values.

/version

Version of the package to be scanned for. The output from the dmsweep packages command describes the valid values. Optional but if you do not supply enough information to uniquely identify the package, an error is displayed.

/platform

Platform property of the package to be scanned for. The output from the dmsweep packages command describes the valid values. Optional but if you do not supply enough information to uniquely identify the package, an error is displayed.

/language

Language property of the package to be scanned for. The output from the dmsweep packages command describes the valid values. Optional but if you do not supply enough information to uniquely identify the package, an error is displayed.

Example:

The following example shows how you can use the dmsweep command line utility. Replace values such as the package number, target computers and manager machines with suitable values.

To scan a target computer to find out if a package is already installed, enter the following command:

```
dmsweep scan /ip <target machine> /pn 2
```

dmsweep sspack

Use this command to display the list of packages found on one or more Scalability Servers.

The command usage is as follows:

```
dmsweep sspack [/manager <manager_machine>]
    [/musername <manager_username> /mpassword <manager_password>]
    [/mfilename <manager_filename>]
    {/ip <address_wild> [/toip <stop_address>] |
    /ipfilename <ip_filename> |
    /domain <domain_name> |             (WinNT only)
    /uri <uniform resource identifier> |
```

```
/targetcred <tc_filename> |
/query <query_name> }
```

Where /manager, /musername, /mpassword and /mfilename are the same as described for 'dmsweep packages (see page 530)'.

Where /ip, /toip, /ipfilename, /targetcred, /domain, /uri and /query are the same as described for 'dmsweep scan (see page 533)'.

/manager

Specifies the name of the manager from which to retrieve the package list. If you omit /manager, the current machine is assumed.

/musername

Specifies the username for the manager computer.

The user will be prompted for the username if /musername is entered without <manager_username>.

/mpassword

Specifies the password for the manager computer.

If <manager_password> is not supplied, dmsweep will prompt the value.

Note: Neither /musername nor /mpassword can be specified without /manager.

/ip

Start address or addresses from which to scan.

You can specify an address as a machine name, if it can be successfully resolved (for example, by DNS), or as an IP address (for example 10.0.0.1). The <address_wild> argument enables you to use asterisks (for example, 10.0.0.* covers the range 10.0.0.1 to 10.0.0.254 and 10.0.*.* covers the range 10.0.0.1 to 10.0.255.254.

You can specify a single address for <address_wild> or a comma-separated list of addresses.

Note: This parameter is the start address only when the /toip option is also used.

/toip

End address for the range to scan.

This option can only be used in conjunction with /ip. When used the scan starts with the address specified by <address_wild> and stops after scanning <stop_address>.

/ipfilename

If you specify /ipfilename, the command reads the list of target machines is from the file <ip_filename>. Each line of the file contains a single target address or multiple addresses separated by commas, for example:

machine1

machine2, machine3, 10.0.0.1

machine5

/domain

If a deployment manager is on a Windows-based computer, you can use /domain to scan the computers in the specified domain.

/uri

If you specify /uri you can use a uniform resource identifier to specify target computers to read from an external directory.

/targetcred

If you specify /targetcred, the command reads the list of target machines and connection credentials from the file <target_credentials_file>.

/query

If /query is specified, the list of targets will be generated when the unmanaged asset query identified by <query_name> is run.

Example:

```
dmsweep sspack /ip ScalServ
Contacting the deployment manager on XXXXX06-CCSTEST.
New job id is 6432350.
Scanning for scalability server packages.
100.100.00.0 ScalServ Found 2 packages (1 of 1)
Scan complete.
SCAN SUMMARY
================================================================================
100.100.00.0 ScalServ Found 2 packages
CA Unicenter DSM Agent + Basic Inventory Plugin 11.1.8160.410 Windows_x86 ENU
CA Unicenter DSM Agent + Software Delivery Plugin 11.1.8160.410 Windows_x86 ENU
================================================================================
```

dmsweep stage

The dmsweep stage command transfers an agent or server package to one or more scalability servers, from where it can subsequently be deployed to target computers using the dmsweep deploy command.

The subcommand has the following format:

```
stage [/manager <manager_machine>]
    [/musername <manager_username> /mpassword <manager_password>]
    [/mfilename <manager_filename>]
    {/ip <address_wild> [/toip <stop_address>] |
    /ipfilename <ip_filename> |
    /targetcred <target_credentials_file> |
    /query <query_name>}
    [/tusername <target_username> /tpassword <target_password> | /epassword
    <encrypted_password>] ]
    [/tfilename <target_filename>]
    {/packagenum <package_number> |
    /product <product_name> /package <package_name>
    [/version <version>] [/platform <platform>] [/language <language>]}
    [/jobname <job_name>]
```

/manager

Specifies the name of the manager from which to retrieve the package list. If you omit /manager, the current machine is assumed.

/musername

Specifies the username for the manager computer.

If <manager_username> is not supplied, dmsweep will prompt for the value.

/mpassword

Specifies the password for the manager computer.

If <manager_password> is not supplied, dmsweep will prompt the value when the command is executed.

/mfilename

Lets you read the name of the manager and details of its username and password from a file.

/ip

Start address or addresses from which to scan.

You can specify an address as a machine name, if it can be successfully resolved (for example, by DNS), or as an IP address (for example 10.0.0.1). The <address_wild> argument enables you to use asterisks (for example, 10.0.0.* covers the range 10.0.0.1 to 10.0.0.254 and 10.0.*.* covers the range 10.0.0.1 to 10.0.255.254.

You can specify a single address for <address_wild> or a comma-separated list of addresses.

/toip

End address for the range to scan.

Has to be used in conjunction with /ip: the scan starts with the address specified by <address_wild> and stops after scanning <stop_address>.

/ipfilename

If you specify /ipfilename, the command reads the list of target machines is from the file <ip_filename>. Each line of the file contains a single target address or multiple addresses separated by commas, for example:

machine1

machine2, machine3, 10.0.0.1

machine5

/targetcred

If you specify /targetcred, the command reads the list of target machines and connection credentials from the file <target_credentials_file>.

/query

If /query is specified, the list of targets will be generated when the unmanaged asset query identified by <query_name> is run.

/packagenum

Package number of the package to be scanned for. The output from the dmsweep packages command describes the valid values.

/product

Property of the package to be scanned for. The output from the dmsweep packages command describes the valid values.

/package

Property of the package to be scanned for. The output from the dmsweep packages command describes the valid values.

/version

Version of the package to be scanned for. The output from the dmsweep packages command describes the valid values. Optional but if you do not supply enough information to uniquely identify the package, an error is displayed.

/platform

Platform property of the package to be scanned for. The output from the dmsweep packages command describes the valid values. Optional but if you do not supply enough information to uniquely identify the package, an error is displayed.

/language

Language property of the package to be scanned for. The output from the dmsweep packages command describes the valid values. Optional but if you do not supply enough information to uniquely identify the package, an error is displayed.

/tusername

Specifies the username for connecting to the target. If your deployment manager has already deployed packages to the target computer, credentials may not be required.

/tpassword

Specifies the password for connecting to the target. If your deployment manager has already deployed packages to the target computer, credentials may not be required.

If you specify /tusername but omit <target_username>, you will be prompted for the username. If you specify /tpassword but omit <target_password>, you will be prompted for the password, which will not be displayed on the screen.

/epassword

Specifies the password in encrypted form. This option alerts dmsweep to this format and the password will be decrypted before being used. See Encrypted Passwords (see page 542).

/tfilename

If a target needs authentication and /tfilename has been specified, <target_filename> is searched. If an entry exists, the credentials specified in the file are used for the target, otherwise /tusername and /tpassword are used. The file <target_filename> contains one target per line, with each line containing the target's machine name (or IP address), username and password, for example:

10.0.0.1 username1 password1

10.0.0.2 username2 password2

* username3 password3

An entry with '*' as the target name denotes default credentials for any target not explicitly listed in the file.

/jobname

Specifies the name of the job is set to the supplied <job_name> argument; otherwise it will be generated automatically based on the current date and time.

Example:

The following example shows how you can use the dmsweep command line utility to stage a package on a scalability server. Replace values such as the package number, target computers and manager machines with suitable values. To stage a package on a deployment scalability server, enter the following command:

```
dmsweep stage /ip <target machine> /pn 2 /tu /tp
```

The user will be prompted to supply both the target username and target password.

dmsweep version

Displays version information for dmsweep.

The subcommand has the following format:

```
version
```

To display the dmsweep version enter the following:

```
dmsweep version
```

Encrypted Passwords

In order to create the encrypted passwords for use with "dmsweep deploy" and "dmsweep stage", you must use the "caf savecreds" command. This command will output to a file called cafcreds.txt in the DSM bin folder, and from there you can copy-and-paste the information into the dmdeploy target credential file or use on the dmsweep command line.

Example

```
caf savecreds "dummy" user "administrator" password "secret" host "machine4"
```

This example results in a cafcreds.txt file containing a line, ‘machine4’ ‘dummy’ ‘ machine4\administrator’ ‘4DDzqKuSJN5Ml6RsEY+o5Q'.

dmsweep Target Credentials File

The target credentials file enables you to specify targets and their passwords in one place. The format of the file is one set of targets per line.

You can use the /ip, /toip, /uri, /domain, /tu, /tp, and /query options in the file, in any order. If a username and password are specified without a target, these become the default username and password; otherwise, there must be at least one target. The default values are used on any line with a target but with no /tu and/or no /tp option. If the /tu or /tp option has no value, the user will be prompted for one. If there is no /tu (or /tp) option, the default username (or default password) is used. A blank password is used where there is no /tp option and no default password.

Note: You may also use the encrypted password option, /ep, in the target credentials file.

Examples of entries in the file are:

```
/ip <machine1,machine2> /tu /tp
/tu defaultUser /tp defauLtPassword
/ip 130.119.3.* /tu administrator /tp pass3472773
/ip 130.119.9.5 /toip 130.119.9.12 /tu administrator /tp
```

The target credentials file allows customers to plan their deployments “offline” before initiating the actual deployment jobs. The files can be generated in response to events and other external factors, and then fed into the deployment process at predetermined times. Deployment to different “areas” of a network could also be specified by creating families of credential files relating to different company departments or offices, for example.

The target credentials file uses syntax similar to the dmsweep command line, which provides consistency and also allows easier expansion to provide additional features (via new option syntax) in future versions.

Credentials File Syntax

Any line in the credentials file starting with ‘#’ is treated as a comment and is ignored.

The Credentials file consists of a number of target specifications, each on a separate line. Each target specification has the following form

```
[<Targets>] [/tu [user]] [/tp [password] | /ep [encrypted password]]
```

The order of elements in a target specification is not significant.

<Targets>

This parameter specifies the target computers to which deployment is to be attempted. It can take one of the following forms:

/ip <address>

A single computer names or IP address.

/ip <address>,<address>, …

A comma separated list of computer names or IP addresses in any combination.

/ip <IP> /toip <IP> [/domain <domain>]

A start and end range of numeric IP addresses. If the domain option is specified, deployment is only attempted to addresses in the IP range which correspond to computers in the named domain. Note that domain evaluation is only supported when using a deployment manager hosted on the Windows platform.

/domain <domain>

A Windows domain, the constituent computers of which are to be the deployment targets. Note that domain evaluation is only supported when using a deployment manager hosted on the Windows platform.

/uri <uri>

A Universal Resource Indicator specifying a directory container, the constituent computers of which are to be the deployment targets.

/query <query_name>

If /query is specified, the list of targets will be generated when the unmanaged asset query identified by <query_name> is run.

/tu [user]

This parameter specifies the target user name. This is the id used to connect to target computers when performing the initial push of DMPrimer. The user id value is applied to all addresses specified by the <targets> element of the target specification. If the user name is omitted, its value will be requested by dmsweep or the deployment wizard when a deployment is attempted using the credentials file.

/tp [password]

This parameter specifies the password of the target user. If the password is omitted, its value will be requested by dmsweep or the deployment wizard when a deployment is attempted using the credentials file. If the /tp option is omitted entirely from a target specification, a default password is used (see below).

/ep [encrypted password]

This parameter specifies the password of the target user in encrypted format. This option alerts dmsweep to this fact and the password will be decrypted before being used.

If <targets> is not specified on a line, the /tu and /tp values specified on that line are treated as default values to be used in all cases where /tu and /tp are omitted on target specification lines lower down the credentials file. It is possible to specify multiple default user & password values in a credentials file, but only the latest ones encountered (reading down the file) are in effect at one time.

Abbreviations for dmsweep Arguments

The following dmsweep arguments are interchangeable:

/domain and /sd

/ip and /a1

/ipfilename and /if

/jobname and /jn

/language and /pl

/manager and /mgr

/mfilename and /mf

/mpassword and /mp

/musername and /mu

/package and /pk

/packagenum and /pn

/platform and /pt

/pparams and /pp

/primerargs and /pri

/product and /pd

/sserver and /ss

/sspassword and /ssp

/ssusername and /ssu

/targetcred and /tc

/tfilename and /tf

/toip and /a2

/tpassword and /tp

/epassword and /ep

/tusername and /tu

/version and /pv

dmdFtpUpload: FTP Upload for Deployment Manager Primers

Valid for Linux

If you use a Linux manager, you can upload the DMPrimers to an FTP server after installation. The dmdFtpUpload script uploads all the DMPrimers it finds in the deployment library for all platforms.

The script is at $CABIN/dmdFtpUpload (or $CA_ITRM_BASEDIR/dmdeploy/scripts/dmdFtpUpload and uses environment variables to control the FTP parameters.

By default, the script uses properties set during installation. You can use the following environment variables to override the installation settings:

DMFTPSERVER

The hostname of the FTP server. The default value is the local host name.

DMFTPUSER

An FTP server account with write privileges. The default value is "anonymous".

DMFTPPWD

Password of $DMFTPUSER. The default value is "root".

DMFTPHOMEDIR

The path on the FTP server, as seen by this client, to which to upload the packages. If the path is relative, it is relative to the $DMFTPUSER account's home directory. For an anonymous account the (ch)root directory is the anonymous account's home directory. The default is "/pub/dmprimer".

Chapter 6: dsmpgm Command - Control Daemon Processes on Linux and UNIX

The command dsmpgm lets you start up, shut down, or query the status of CA ITCM daemon processes on Linux and UNIX. It is essentially a wrapper to the caf command, which is the primary command line interface to control CA ITCM daemons. However, dsmpgm ensures that external prerequisite processes are available prior to invoking the caf functionality.

The command has the following format:

```
dsmpgm {start | restart | stop | status}
```

start

Start the daemon processes.

stop

Stop the daemon processes.

restart

Stop and start the daemon processes.

status

Query the status of the daemon processes.

Exit status:

Displays the status of the command execution. The value zero (0) means OK, any non-zero value indicates Error.

Chapter 7: The dsmPush Tool

The dsmPush tool is used to import (or "push") and create CA ITCM component installation packages from the installation DVD into the domain manager. The dsmPush tool can be used to import packages suitable for use by either Infrastructure Deployment or Software Delivery.

Language independent base packages are provided on the installation DVD for the Basic Hardware Inventory (BHI) agent, Asset Management (AM) agent, Remote Control (RC) agent, and Software Delivery (SD) agent. The language independent base packages already include the English (ENU) language package; therefore, there is no separate English (ENU) language package.

You can create your own set of agent packages using the dsmPush tool. You can specify which agent combination and which languages you want. For example, you can create a package containing Software Delivery and Remote Control with the Spanish and Korean language packs.

For further details, see the “About Agent Language Package Creation and Installation” section in the *Implementation Guide*.

As part of setup a predefined set of packages are pushed onto the CA ITCM manager. With the dsmPush tool you can push these packages again or create new packages to push.

The dsmPush tool is written using CA ITCM scripting language and provides a check function to validate and list existing packages in the CA ITCM manager, and a copy function which performs the push or creation of the packages into the CA ITCM manager.

On Windows, dsmPush is called as follows:

`dmscript.exe` *`pathname`*`\dsmPush.dms {copy | check} [-I imagelocation] [`*`parameters`*` ...]`

`The parameter -? shows a "Usage" to the user.`

dsmPush -? - Usage -

The parameter -? shows a "Usage" to the user. The usage is also shown if dsmPush is called without parameters, or if there is a severe parameter error. "Usage" is also controlled by the parameter -silent.

```
dmscript.exe pathname\dsmPush.dms { check <check-parameters> | -? |
    copy [-I imagelocation] [<copy-parameters>] [-silent] }

<copy-parameters> =
    { [-single] [<os-parameters>] [<sd/dm-Parameters>] }
    [<pack-parameters>] [<lang-parameters>]

<os-parameters>     =    [-win] [-hp] [-sun] [-ibm] [-linux] [mac] [unixware]

<sd/dm-Parameters>=      [-REGSD] [-REGDM]

<pack-parameters> =      [-rc] [-sd] [-am]

<lang-parameters> =
    [-P nls,enu,chs,deu,esn,fra,jpn,kor]
    [-L nls,enu,chs,deu,esn,fra,jpn,kor]

<check-parameters> = [-f check-output-file-name]
```

dsmPush Check Function

```
dsmPush check
```

Checks whether deployment packages exist at the local system and lists the packages found. The tool dsmPush does not check for Software Delivery packages. The list of packages found by dsmPush contains the following information:

- Product Name
- Package
- Operating environment
- Language
- Version
- Description

The information is written to the check-output-file. Its name defaults to DSM_dsmPush_check_output.txt.

On Windows its location defaults to the temporary folder (%TEMP%)

On Linux its location defaults to $CA_ITRM_BASEDIR/logs if that variable is set or /tmp if not.

dsmPush copy script—Import Packages into a Library

The dsmPush copy script imports a set of packages for the specified products and operating environments into the Infrastructure Deployment library on the local system or into the Software Delivery library.

Note: To see the usage of the dsmPush script, issue a single question mark as parameter with the dsmPush script. The script must be run from the CD image, as follows:

- On Windows:
  ```
  dmscript dsmPush -?
  ```
- On Linux:
  ```
  dmscript dsmPush.dms -?
  ```

On Windows, the usage is displayed in a popup message box; on Linux, the usage is printed to standard output (stdout). The usage displays automatically in the case of parameter errors; however, the usage is not displayed when the -silent parameter is used with dsmPush.

-I "*imagelocation*"

Specifies the root directory of the image. The default is the location of the dsmPush.dms script.

operating_environment

Specifies the operating environment of the packages to import through one of the following options:

-win	Windows platforms
-linux	Linux platforms
-sun	Sun Solaris platforms
-ibm	IBM AIX platforms
-hp	HP UX platform
-mac	MAC OS X platforms
-unixware	SCO UnixWare platforms

product

Specifies the combination of products each imported agent package should contain. Valid values are as follows:

-am	Asset Management
-rc	Remote Control
-sd	Software Delivery

registration

Specifies to which library the packages should be imported. Valid values are as follows:

-REGDM	Specifies that the packages should be imported into the Infrastructure Deployment library on the local system
-REGSD	Specifies that the packages should be imported into the Software Delivery library.

If nothing is specified for *registration*, the packages are imported into both libraries.

-P {*list_of_languagepackages*}

Specifies the language packages to import.

The *list_of_languagepackages* is a comma-separated list without any spaces where you can specify the language packages using the following options:

nls	multi-language packages
enu	English (U.S.) packages
chs	Simplified Chinese packages
deu	German packages
esn	Spanish packages
fra	French packages
jpn	Japanese packages

kor Korean packages

Notes:

This is the complete list of supported languages at the time of publication but more may be added later.

The nls is provided for compatibility purposes only. It should be used only in combination with enu. The nls option represents the three languages French (fra), German (deu), and Japanese (jpn).

When you specify nls, an old style NLS package is copied to the library (Windows only!).

Examples:

```
-P deu,fra,kor
```

imports packages that can be configured to operate in any of German, French, and Korean language.

```
-P enu
```

specifies that only the English-only base packages are copied.

-L {enu|chs|deu|esn|fra|jpn|kor}

Specifies the language used by the Infrastructure Deployment wizard and the Software Delivery registration information. This is usually the language in which the CA ITCM domain manager is running. You can use one of the options listed to specify the language. If the -L parameter is missing, enu (English (U.S.)) is used by default.

Example: -L deu

-single

Forces that only one package (agent or language package) is imported in the Software Delivery and Infrastructure Deployment libraries. Any other additional package that is usually also imported is omitted. This applies to each operating environment specified or for all operating environments if the *operating_environment* parameter is missing.

So if more than one operating environment is specified the result will not be a single package but a single package per operating environment

Examples:

-single -P chs

Imports the Chinese language package.

-single -am

Imports the Asset Management agent package (English only).

-single -am -rc -P deu,fra

Imports the merged Remote Control and Asset Management agents package for English, German, and French.

-silent

Switches dsmPush's progress report off (for backward compatibility reasons).

By default, dsmPush provides a progress report that informs the users about what has already been done. The first information in the report is the name of the log file. On Linux, the progress report is written to standard output (stdout); on Windows, it is written to the Script Interpreter Window. This window closes 10 seconds after the progress report script has finished.

Using Combinations of Parameters

Following are some important notes on using combinations of the parameters described above:

- If the *operating_environment* parameter is not specified, the specified packages are imported for all supported operating environments, independent of the -single option.
- If the *product* parameter is not specified and the -single option is missing, all agent packages available on the CD for the languages specified through the -P parameter are imported. If the -single option is specified, the specified language package is installed.
- If no language is specified (-P parameter is missing), then for all specified packages only the NLS version (= enu,deu,fra,jpn) is imported.
- If the -single option is specified but the *product* and -P parameters are missing, no package will be imported.

More examples for the usage of the dsmPush script on Windows and Linux to import packages in the Software Delivery and Infrastructure Deployment libraries can be found in section "dsmPush Examples".

dsmPush Examples

Display the dsmPush Usage

From the top level DVD folder run the dsmPush script, for example:

```
C:\DSM_11_2_226_1042_CD> dmscript dsmPush -?
```

Import Specific Product Packages with Language Packages Using the German Version of the Infrastructure Deployment Wizard and the Software Delivery reginfo Files

This example imports Asset Management and Software Delivery product packages with German and French language packages in the libraries, using the German version of the Infrastructure Deployment wizard and the Software Delivery reginfo files. The example applies to a Windows environment:

```
...\Program Files\CA\DSM\Bin\dmscript.exe "pathname\dsmPush.dms" copy
-I "imagelocation" -am -sd -REGDM -REGSD -P enu,deu,fra -L deu
```

Import Windows Packages to Support All Languages

```
dmscript c:\DSM_11_2_226_1042_CD\dsmPush.dms copy
-I c:\DSM_11_2_226_1042_CD -win -P fra,chs,deu,esn,kor,jpn -L enu
```

Import Windows Packages for Asset Management in French Language to the Software Delivery and Infrastructure Deployment Libraries

```
dmscript c:\DSM_11_2_226_1042_CD\dsmPush.dms copy
-I c:\DSM_11_2_226_1042_CD -win -am -P fra -L enu
```

Import Windows Packages for Remote Control in Spanish Language to the Software Delivery and Infrastructure Deployment Libraries

```
dmscript c:\DSM_11_2_226_1042_CD\dsmPush.dms copy
-I c:\DSM_11_2_226_1042_CD -win -rc -P esn -L enu
```

Import Windows Packages for Remote Control and Software Delivery in Chinese Plus Japanese Language to the Software Delivery and Infrastructure Deployment Libraries

```
dmscript c:\DSM_11_2_226_1042_CD\dsmPush.dms copy
-I c:\DSM_11_2_226_1042_CD -win -rc -sd -P chs,jpn -L enu
```

Import Windows Packages for Asset Management and Software Delivery in Chinese, Korean, German, and Japanese Language to the Software Delivery and Infrastructure Deployment Libraries

```
dmscript c:\DSM_11_2_226_1042_CD\dsmPush.dms copy
-I c:\DSM_11_2_226_1042_CD -win -am -sd -P kor,jpn,chs,deu -L enu
```

Import Linux Packages for Asset Management and Software Delivery in Chinese, Korean, German, and Japanese Language to the Software Delivery and Infrastructure Deployment Libraries

```
dmscript c:\DSM_11_2_226_1042_CD\dsmPush.dms copy
-I c:\DSM_11_2_226_1042_CD -Linux -am -sd -P kor,jpn,chs,deu -L enu
```

Import All Windows Packages in All Languages to the Software Delivery Library

```
dmscript c:\DSM_11_2_226_1042_CD\dsmPush.dms copy
-I c:\DSM_11_2_226_1042_CD -win -REGSD -P kor,jpn,chs,deu,esn,fra -L enu
```

Import All Windows Packages in All Languages to the Infrastructure Deployment Library

```
dmscript c:\DSM_11_2_226_1042_CD\dsmPush.dms copy
-I c:\DSM_11_2_226_1042_CD -win -REGDM -P kor,jpn,chs,deu,esn,fra -L enu
```

Import All Windows Packages in All Languages to the Software Delivery and Infrastructure Deployment Libraries Unattended

```
dmscript c:\DSM_11_2_226_1042_CD\dsmPush.dms copy
-I c:\DSM_11_2_226_1042_CD -win -P fra,chs,deu,esn,kor,jpn,nls -L enu
-silent
```

Chapter 8: intellisigcmd - Command Line Tool

intellisigcmd is a command line tool for Intellisigs. This tool has the following format:

```
intellisigcmd <cmd> param1=value1 param2=value2 ... [<DB_Credentials>]
```

cmd

Specifies the import, export, or genuuid command.

DB_credentials

Specifies the database credentials of the MDB. By default, the credentials are retrieved from the comstore.

Use the following sample format to specify the DB credentials:

Example: SQLServer DB Credentials format

```
dbvendor=mssql dbhost=myhost dbname=mdb dbuser=ca_itrm dbpassword=mypwd
dbinstance=inst
```

Example: Oracle DB Credentials format

```
dbvendor=oracle dbhost=myhost dbname=orcl dbuser=ca_itrm dbpassword=mypwd
dbinstance=1521
```

intellisigcmd export—Export Intellisigs

The intellisigcmd export command lets you export Intellisigs.You can either use the DSM Explorer or use the command to export Intellisigs.

This command has the following format:

```
intellisigcmd export file=<export name> [type=xml|zip] [platform=all|windows|unix]
```

export name

Specifies the name of the Intellisig XML or zip file that you want to export. If you do not provide the file extension, and the type is xml, the command creates a folder with the supplied name.

type

Specifies whether you want to export an XML or zip file. If you do not include the type parameter, the command assumes the export type depending on the export file extension.

Valid values: xml, zip

platform

Specifies the platform to determine which Intellisigs are exported.

Valid values: all, windows, unix

Default: all

intellisigcmd import—Import Intellisigs

The intellisigcmd import command lets you import Intellisigs.You can either use the DSM Explorer or use the command to import Intellisigs.

This command has the following format:

```
intellisigcmd import [file=<import source>] [type=xml|zip]
[mode=replace|mergenew|mergeall] [updateactive=yes|no] [delete=yes|no]
```

import source

Specifies the name of the XML or zip file to which you want to import the Intellisig. If you do not provide the file extension, the command assumes the file extension depending on the type.

Note: If you want to import to an XML file, verify that the supporting directories exist in the same folder as the XML file.

type

Specifies whether you want to import as XML or zip file. If you do not include this parameter, the command assumes the import type depending on the import file extension.

Valid values: xml, zip

mode

Specifies the import mode. Following import modes are supported:

Default: mergenew

replace

Replaces existing definitions with the definition being imported. Existing definitions are lost.

mergenew

Appends new Intellisig versions to the definitions on the manager. Existing definitions are not modified.

mergeall

Appends new Intellisig versions and updates the existing definitions that are included in the import file. Intellig versions that are not defined in the import files are not modified.

updateactive

Specifies whether active Intellisig versions can be updated during the import.

Valid values: Yes, Y, true, 1 or No, N, false, 0

Default: Yes

delete

Specifies whether you want to delete Intellisigs before the import. If you do not include the delete switch, none of the Intellisigs are deleted before import.

Default: No

intellisigcmd genuuid—Generate UUIDs

The intellisigcmd genuuid command lets you generate unique UUIDs which you can use when creating custom Intellisigs.

This command has the following format:

```
intellisigcmd genuuid [num=<count>]
```

num

Specifies the number of UUIDs to be generated. If you do not specify this parameter, a single UUID is generated. Otherwise, <count> UUIDs are generated.

Valid Values: 1 to 1000

Chapter 9: MDB Admin Console CLI

MDB Admin Console is a command line interface (CLI) for exporting and importing data related to CA IT Client Manager from one MDB to the other. For example, you can export data from your test environment and import it into your production environment using MDB Admin Console.

This section contains the following topics:

MDB Admin Console

The MDB Admin Console lets you deploy CA ITCM data from a source MDB to a target MDB and perform certain database administration tasks. CA ITCM exposes the MDB Admin Console as a command line interface (CLI).

You can use the MDB Admin Console to do the following tasks:

- Export and import data related to CA ITCM from one MDB to the other. For example, you can export data from your test environment and import it into your production environment using MDB Admin Console.
- Reconfigure database entries to adapt to a change in the network (IP) name of the DSM manager.
- Reconfigure database entries to adapt to a change in the network (IP) name of the DBMS manager.

Export and Import Considerations

The following considerations apply when you use MDB Admin Console to export or import data:

- MDB Admin Console can export and import the following MDB objects:
  - Security Profiles and Area definitions
  - Asset Groups
  - Queries
  - Query-based and Event-based Asset Policies
  - Configuration Policies
  - Software Groups, Procedure Groups, and Software Policies
  - Software Definitions

  Note: For a detailed description of the supported MDB objects, see the "Supported MDB Object Types" appendix.
- You can perform export and import as two separate tasks. This approach lets you review the exported MDB data, which is stored in XML files, before you import the data into the target MDB.

  Note: Do not change the contents of the XML file if you are planning to use the XML file for importing data into the target MDB later.
- MDB Admin Console does not delete any object from the target MDB when you import data.

Exporting and Importing Data Across MDB Platforms and Version

You can use MDB Admin Console to export and import data in the following scenarios:

CA ITCM r12.5, or r12.8 to equivalent or higher versions of CA ITCM

You can export the MDB Objects from a CA ITCM r12.5, or r12.8 MDB and import them into the MDB of an equivalent or higher version of CA ITCM. For example, you can export the MDB objects from a CA ITCM r12.5 MDB and import them into a CA ITCM r12.8 MDB.

Enterprise to Enterprise

You can export the MDB Objects from a DSM enterprise manager MDB and import them into another CA ITCM enterprise manager MDB.

Domain to Domain

You can export the MDB Objects from one domain manager MDB and import them into another domain manager MDB.

SQL Server to Oracle

You can export the MDB Objects from a SQL Server MDB and import them into an Oracle MDB. Similarly, you can export the MDB Objects from an Oracle MDB and import them into a SQL Server MDB.

Note: You can combine the export and import scenarios. For example, you can export the MDB Objects from a CA ITCM r12.5 enterprise manager MDB and import into a CA ITCM r12.8 enterprise manager MDB.

MDB Admin Console Files and Folders

The installer creates the following files and folders related to MDB Admin Console upon successful installation:

- **.\mdbac.bat** - Executable batch file which exposes the command line interface.
- **.\log** – Folder which includes log files from the MDB Admin Console. For more details about the log folder, see the Troubleshooting section.
- **.\cfg –** Folder including configuration files.
- **.\sample*.bat** – There are a number of files that you can use as examples of how the MDB Admin Console command line can be used.

Configure Database Connections

MDB Admin Console uses connection names to reference the MDB databases for its operations. Therefore, before using the MDB Admin Console, you need to define the connection names for the MDBs involved in the export and import operation. You supply these connection names as an argument to the MDB Admin Console command line interface.

Follow these steps:

1. Open the mdbac.properties file located in the .\cfg folder. The .\cfg folder is available in the MDB Admin Console installation folder.
2. Search for "Define the Connections" and either modify the existing database connection entries or add new entries for new database connections. Define the new database connection strings in the following format:

   Oracle MDB

   `connection.`*`connection_name`*`=jdbc:oracle:thin:@`*`Host`*`:`*`Port`*`:`*`SID`*

   SQL Server MDB

   `connection.`*`connection_name`*`=jdbc:sqlserver://`*`Host`*`\[`*`Instance_name`*`]:`*`Port`*`;DatabaseName=`*`database_name`*

   To configure a trusted connection to the SQL Server MDB, use the following format:

   `connection.mssql-test`
   `=jdbc:sqlserver://`*`Host`*`\[`*`instance_name`*`]:`*`Port`*`;DatabaseName=`*`database_name`*`;integratedSecurity=true`

   Connection_name

   Defines the connection name. You specify this connection name as a parameter to the MDB Admin Console CLI to refer to the MDB.

   Host

   Defines the name of the computer that hosts the MDB.

   Instance_name

   (Optional) Defines the instance name to connect to.

   Default: Default instance

   Note: If your instance name starts with 'n', 'r', or 't', you must add the escape sign '\' before the character because these characters act as terminators when prefixed with a '\'. For example, if the instance name is nonname, the connection string must be as follows:

   `MyMachine\\nonname:1488`

   Note: The instance_name parameter is mandatory for the adjustmdbcfg method even if you are using the default instance. The default instance name must be provided as "MSQLSERVER".

Port

Defines the port number of the MDB.

Database_name

Defines the database name of the MDB.

SID

Defines the SID of the MDB.

IntegratedSecurity=true

Indicates that trusted connection to the SQL Server MDB is required. You need a trusted connection to the SQL Server MDB for running the adjustmdbcfg method (see page 573).

Examples:

```
connection.mssql-test =jdbc:sqlserver://lab_test:1433;DatabaseName=mdb
```

```
connection.mssql-test
=jdbc:sqlserver://lab_test\\MSQLSERVER:1433;DatabaseName=mdb;integratedSecuri
ty=true
Connection.oracle-test=jdbc:oracle:thin:@lab-test:1521:orcl
```

3. Verify that the database connection to the MDB Admin Console is working. To do this, follow these steps:

   a. Open the command prompt and navigate to the MDB Admin Console installation directory using the following command:

      ```
      cd MDBAC_Installation_Directory
      ```

   b. Execute the following command:

      ```
      mdbac –testcon –target Connection_name -user username –pwd password
      ```

      Verifies the database connection and returns the result.

      Examples:

      ```
      mdbac –testcon –target mssql-test -user ca_itrm –pwd password123
      ```

      ```
      mdbac –testcon –target oracle-test -user mdbadmin –pwd password123
      ```

   You can now use the connection names in your MDB Admin Console CLI.

More information:

MDB Admin Console CLI (see page 567)

Chapter 10: Using MDB Admin Console

Command Line Arguments

The following command line arguments are common to all or most of the MDB Admin Console commands:

user

Specifies the user name for connecting to the database. The default user name is ca_itrm for Microsoft SQL Server MDB and mdbadmin for Oracle MDB.

pwd

Specifies the password for connecting to the database.

wallet

Specifies the path to the wallet file that contains the user name and password for connecting to the database. Use this argument as an alternate to the user and pwd arguments. Before passing the wallet argument to the MDB Admin Console CLI, you must create the file that contains the credentials using the createwallet method.

Work_folder

Specifies the relative or absolute path to a folder for storing data such as, exported xml data files. You need to verify that the folder exists in the specified path.

Connection name

Specifies the connection name that you defined while configuring the database connections in the mdbac.properties file.

More information:

createwallet method—Create a Wallet (see page 568)

MDB Admin Console CLI

This section describes the methods that MDB Admin Console provides for various database administration tasks.

Note: To execute the MDB Admin Console CLI, change directories to the MDB Admin Console installation folder in the command prompt.

Order of Exporting and Importing MDB Objects

When you use MDB Admin Console to export MDB objects from a source MDB and to import them into a target MDB, you must follow the order below:

1. Security profiles
2. Areas
3. Queries
4. Asset groups
5. Query-based policies
6. Event-based policies
7. Software groups
8. Procedure groups
9. Software-based policies
10. Software definitions
11. Configuration policies

createwallet method—Create a Wallet

A wallet is a file that contains the user name and password for connecting to the database. You can provide a wallet as an argument to the MDB Admin Console CLI.

This command has the following format:

```
mdbac.bat –createwallet
```

This command prompts you with a dialog to enter the user name and password for connecting to the database. Click Save Wallet to save the credentials to a file. If the specified file already exists, clicking Save Wallet overwrites the file.

export method—Export MDB Objects

The export method exports MDB objects from a source MDB. It creates an XML file for each object that matches the specified object name or pattern. The XML files are named as follows:

ObjectType__ObjectName__(*counter*).xml

Example: XML file name

BoQueryDef__Asset is a Virtual Machine - MS Virtual PC__(1).xml

This command has the following format:

```
mdbac.bat –export MDB_Object_Type -name Object_Name - work workfolder -source
Connection_Name {–user User_Name –pwd Password | -wallet wallet_file_name}
```

MDB_Object_Types

Specifies the MDB object type that you want to export.

Note: For a list of supported MDB object types, see the "Supported MDB Object Types" appendix.

Object_Name

Specifies the object name in the given object type that you want to export. You can use wildcard characters to match a pattern. For example, you can specify "Asset%" to process all the objects that start with the name "Asset" in the given object type.

Example: Export MDB Objects

This example command exports queries that start with the object name "*Asset is a Virtual Machine"* from a specified SQL MDB:

```
mdbac.bat -export BoQueryDef -name "Asset is a Virtual Machine%" -work C:\TEMP\work
-source mssql -user ca_itrm -pwd password123
```

Example: Using the Wallet Argument

```
mdbac.bat -export BoQueryDef -name "Asset is a Virtual Machine%" -work C:\TEMP\work
-source mssql -wallet dbwallet
```

import method—Import MDB Objects

The import method imports MDB Objects into a target MDB. It creates new objects and updates existing objects in the target MDB. For each of the supported MDB Object types, the object name is used as the unique reference to identify whether the object is new or existing.

Note: If you do not find the imported objects in the DSM Explorer, restart CAF.

This command has the following format:

`mdbac.bat –import –file file_or_folder_name -target Connection_Name {–user User_Name –pwd Password | -wallet wallet_file_name}`

Note: While you can use the ca_itrm user credentials to import data into SQL Server MDB, you need the credentials of the mdbadmin user to import data into Oracle MDB.

File_or_Folder_Name

Specifies the name of the file or folder that contains the exported data.

Note: You can use the import method to import multiple files at one time. However, avoid storing several files of the same MDB object type and object name in one folder because the sequence of processing these files and the result of the import method cannot be determined.

Example: Import MDB Objects

This example command imports MDB objects from the work folder to the target MDB:

```
mdbac.bat –import -file C:\TEMP\work -target mssql -user ca_itrm -pwd password123
```

compare method—Compare MDB Objects in Source and Target MDBs

The compare method compares MDB Objects in the source and target MDBs. It reads objects from source and target databases and creates corresponding xml files in the work folder. Each supported MDB object type has specific rules that determine when the objects are considered equal. For a list of these rules, see the "Rules for MDB Object Comparison" appendix.

Note: The source and target MDBs must have the same database credentials.

This command has the following format:

```
mdbac.bat -compare MDB_Object_Type -name Object_Name - work work_folder -source
Connection_Name –target Connection_Name {–user User_Name –pwd Password | -wallet
wallet_file_name}
```

MDB_Object_Type

Specifies the MDB object type that you want to compare. This parameter accepts only a single object type. If you include multiple object types, the command returns invalid results.

Note: For a list of supported MDB object types, see the "Supported MDB Object Types" appendix.

Object_Name

Specifies the object name in the given object type that you want to compare. You can use wildcard characters to match a pattern. For example, you can specify "Asset%" to process all the objects that start with the name "Asset" in the given object type.

Example: Compare MDB Objects in Source and Target MDBs

This example command compares the queries that start with "Asset is a Virtual Machine" in test and production databases:

```
mdbac.bat –compare BoQueryDef -name "Asset is a Virtual Machines" -work C:\TEMP\work
-source mssql-test - target mssql-production -user ca_itrm -pwd password123
```

Note: For more details about why the compare command failed, you can view the diff.log file. The diff.log is available under the folder given in the -work parameter of the compare command.

view method—View Properties of an Exported MDB Object

The view method prints the properties of an exported MDB object (contained in an XML file) on the console. It reads the properties from the XML file and prints them on the console. The property names that the export method has generated need not necessarily match the property names in DSM Explorer.

This command has the following format:

```
mdbac.bat –view –file file_name
```

File_name

Specifies the name of a single file, the properties which you want to view.

Example: View Properties of an Exported MDB Object

This example command reads the properties from the BoQueryDef__Asset is a Virtual Machine - VMWare__(0).xml file and prints them to the console:

```
mdbac.bat -view –file C:\TEMP\work\BoQueryDef__Asset is a Virtual Machine - VMWare__(0).xml
```

unregman method—Unregister a DSM Manager

The unregman method unregisters a DSM manager from an MDB. After you have unregistered a DSM manager, you can install a new DSM manager and point it to this MDB. On the first startup, the new DSM manager registers with the MDB you pointed it to. The DSM domain and enterprise managers allow registrations with the corresponding domain and enterprise MDBs only. For example, you can register a domain manager only with a domain MDB.

Note: Before executing the command, verify that target MDB does not have any open sessions.

This command has the following format:

```
mdbac.bat –unregman -target connection_name {-user User_Name -pwd password | -wallet wallet_file_name}
```

Example: Unregister a DSM Manager

This example command unregisters the DSM manager associated with the test MDB:

```
mdbac.bat –unregman -target mssql-test -user ca_itrm -pwd password123
```

adjustmdbcfg method—Configure a DSM Manager to Use a Different MDB

The adjustmdbcfg method reconfigures a DSM manager to use a different MDB. You need to verify that the new MDB is not associated with any other DSM manager.

You must consider the following factors when you are using the adjustmdbcfg method:

- This method is available only for SQL Server MDB.
- To execute this method, install MDB Admin Console on the DSM manager that you want to reconfigure.
- The method lets you reconfigure the components of only the DSM manager. It does not include any reconfiguration for other components like CCS or DTS.
- The method allows reconfiguration of domain and enterprise managers to corresponding domain and enterprise MDBs only. For example, you can reconfigure a domain manager to point to a domain MDB only.
- Verify that the target MDB is accessible using a trusted connection.

The command for this method has the following format:

```
mdbac.bat –adjustmdbcfg -target connection_name –file rules_properties_file
```

Note: The connection string of the connection name that you are using for this method must explicitly state the database instance name even if you use the default instance. For more information, see Configure Database Connections (see page 564).

file

Specifies the path to the rules properties file that contains the reconfiguration steps and actions. For more information, see Rules Properties File (see page 573).

Upon execution, this command updates the following MDB connection parameters in the comstore of the DSM manager:

- Host
- Port
- Database Name

Note: If you have configured the rules properties file to create the ca_itrm user in the target MDB, you must verify that the password of the ca_itrm user in the comstore and in the rules properties file match. This is because the adjustmdbcfg method does not update the ca_itrm user and password parameters in comstore.

Rules Properties File

The rules properties file is a configuration file that lets you configure the steps that adjustmdbcfg method performs for an MDB move. An example file, mdbmove.properties, is available in the *MDB_Install_Directory*\cfg folder.

The following illustration displays the contents of the example file:

```
# step to be executed during a reconfiguration step
#

# syntax
#
# step<x>.action   { createlogin | droplogin | repairlogin
#                                    updatecomstore | updatedomain }
#
# parameters for createlogin
#       step<x>.arg1.name       - mandatory
#       step<x>.arg2.pwd        - mandatory
#
# parameters for droplogin
#       step<x>.arg1.name       - mandatory
#
#
# parameters for repairlogin
#       step<x>.arg1.name       - mandatory
#
# parameters for updatecomstore
#       <none>
#
# parameters for updatedomain
#       step<x>.arg1.label      - optional, default value is server name

reconfigure.sequence=step1,step2,step3,step4,step5

# create a login
step1.action=createlogin
step1.arg1.name=testuser1
step1.arg2.pwd=testuser

# align login to exiting dbuser
step2.action=repairlogin
step2.arg1.name=testuser1


# create a login
step3.action=createlogin
step3.arg1.name=ca_itrm_ams
step3.arg2.pwd=NOT_changedR11

# align login to exiting dbuser
step4.action=repairlogin
step4.arg1.name=ca_itrm_ams

# update coomstore
step5.action=updatecomstore

# update domain
step6.action=updatedomain
step6.arg1.label="new domain name goes here"
```

The following parameters and actions are available in the rules properties file:

reconfigure.sequence

Specifies the sequence of steps that the adjustmdbcfg method must follow. The sequence contains the step names in a comma separated list. Each step has an action name and parameters that are defined in the following format:

`step`*`n`*`.action`

`stepn.arg`*`m`*

createlogin (*login_name, password*)

Creates a database login with the given user name and password in the SQL Server MDB.

droplogin (*login_name*)

Drops the given database login from the SQL Server MDB.

repairlogin (*login_name*)

Aligns the SID of the SQL Server login with the user of the MDB.

updatecomstore

Update the comstore file of the local DSM manager with the database connection properties of the current SQL Server MDB (host name, instance name, port).

updatedomain([label])

Updates the ca_n_tier table of the MDB according to the properties of the server hosting the SQL Server MDB. If you have specified a label, it is used as the label for the domain.

testcon method—Test Database Connection

The testcon method tries to connect to the target database with the specified user name and password, and writes success or failure message to the console.

This command has the following format:

```
mdbac.bat –testcon –target Connection_name {–user User_name –pwd password | -wallet wallet_file_name}
```

Example: Test Database Connection

This example command tests the connection to the target database—mssql-test:

```
mdbac.bat –testcon –target mssql-test –user ca_itrm –pwd password123
```

createschema method—Create Clone Tables

The createschema method creates clone tables and procedures in the target database. The import method calls the createschema method in turn and creates the clone tables and procedures automatically. However, you can call the createschema method manually, if there is a problem with importing the data.

This command has the following format:

```
mdbac.bat –createschema -target connection_name {–user user_name –pwd password -wallet wallet_file_name}
```

Example: Create Clone Tables

This example command creates clone tables in the target database—mssql-test:

```
mdbac.bat –createschema -target mssql-test –user ca_itrm –pwd password123
```

dropschema method—Drop Clone Tables

The dropschema method drops the clone tables and procedures, created by the import method, from the target database. The import method creates clone tables for the import operation. These are temporary tables and you can drop these tables after the import operation is complete.

This command has the following format:

```
mdbac.bat –dropschema -target connection_name {–user user_name –pwd password |
-wallet wallet_file_name}
```

Example: Drop Clone Tables

This example command drops clone tables in the target database—mssql-test:

```
mdbac.bat –dropschema -target mssql-test –user ca_itrm –pwd password123
```

Chapter 11: Troubleshooting

Error Logging

MDB Admin Console uses Log4J for creating the log files. By default, MDB Admin Console log files are created in a folder named log. The log folder is created as a subfolder under your current working folder. The default log level is ERROR. For each call of the mdbac.bat command, a separate log file is created.

Configure Logging

You can configure Log4J to modify the location of the log files or the log level.

To configure logging, edit the *Installation_folder*\cfg\log4j.properties file and specify the log file location or the log level.

The following snippet shows the contents of the log4j.properties file:

```
#*****************************************************************************
#*****************************************************************************
# configure "MDB Admin Console" FILE logger
#*****************************************************************************
#*****************************************************************************
# Configure FILE log appender
log4j.appender.FILE = org.apache.log4j.FileAppender
log4j.appender.FILE.File =${user.dir}/log/mdbac_${log4j.dateTime}.log
log4j.appender.FILE.Append=true
….
#*****************************************************************************
#*****************************************************************************
# mdbac logging options
#*****************************************************************************
#*****************************************************************************
log4j.logger.ca.itcm.mdbac.api=ERROR, FILE
log4j.logger.ca.itcm.mdbac.console=ERROR, FILE
log4j.logger.ca.itcm.mdbac.cmd.ccnfregdb=ERROR, FILE
```

Error Messages

MDB Admin Console returns the following error messages when it is unable to process a request:

MDBAC-0=[MDBAC-0] no error detected

MDBAC-1=general command line error.

MDBAC-2=command line error

MDBAC-3=business class name unknown

MDBAC-4=no working folder given

MDBAC-5=internal error: the business class does not support the IBOExport interface

MDBAC-6=configuration/connection not found

MDBAC-7=input file not found

MDBAC-101=[MDBAC-101] feature is not supported at this level of implementation

MDBAC-102=[MDBAC-102] internal error: no import interfaces found

MDBAC-103=[MDBAC-103] unable to create database session

MDBAC-104=[MDBAC-104] cannot create Object

MDBAC-105=[MDBAC-105] Stored procedure for pushing data failed

MDBAC-106=[MDBAC-106] constrain violation during import

MDBAC-107=[MDBAC-107] verification of target schema failed

MDBAC-108=[MDBAC-108] failed to generate transaction id

MDBAC-109=[MDBAC-109] illegal XML file format detected during import/view

MDBAC-110=[MDBAC-110] failed to read name ids form the database

MDBAC-111=[MDBAC-111] failed to transform business object

MDBAC-112=[MDBAC-112] failed to import inventory name ids

MDBAC-113=[MDBAC-113] one or more files are failed during import of a folder

MDBAC-114=[MDBAC-114] cannot create connection to source database

MDBAC-115=[MDBAC-115] cannot create connection to source database

MDBAC-116=[MDBAC-116] connection name not found

MDBAC-117=[MDBAC-117] database type no supported

MDBAC-118=[MDBAC-118] illegal connection configuration, failed to setup a configuration to connect to the database

MDBAC-119=[MDBAC-119] error detected when reading the data from the database

MDBAC-120=[MDBAC-120] not all objects are exported, one or more object failed to get it exported

MDBAC-121=[MDBAC-121] deletion of one or more business objects failed

MDBAC-125=[MDBAC-125] database connection does not exists

MDBAC-126=[MDBAC-126] failed to adjust MDB configuration.

MDBAC-127=[MDBAC-127] connection name not found in properties file containing the adjust rule

MDBAC-128=[MDBAC-128] failed to adjust manager with MDB

MDBAC-129=[MDBAC-129] adjust properties file does not exist

MDBAC-201=[MDBAC-201] missing one or more command line arguments

MDBAC-202=[MDBAC-202] configuration undefined

MDBAC-203=[MDBAC-203] configuration file/name not found

MDBAC-204=[MDBAC-204] missing SQL file

MDBAC-205=[MDBAC-205] SQL folder does not exist

MDBAC-206=[MDBAC-206] SQL file does not exist

MDBAC-207=[MDBAC-207] missing name of database user

MDBAC-208=[MDBAC-208] missing password of the database user

MDBAC-209=[MDBAC-209] cannot connect to database

MDBAC-210=[MDBAC-210] command aborted

MDBAC-211=[MDBAC-211] error in SQL script

MDBAC-212=[MDBAC-212] creation of tables failed to extend the target schema

MDBAC-213=[MDBAC-213] failed to delete the target schema extension

MDBAC-214=[MDBAC-214] failed to verify the target schema extension

MDBAC-215=[MDBAC-215] failed to align the collation settings of the target schema extension

MDBAC-216=[MDBAC-216] failed to create target schema

MDBAC-217=[MDBAC-217] failed to align target schema, schema configuration problem

MDBAC-218=[MDBAC-218] session exists but failed to read connection information

MDBAC-219=[MDBAC-219] failed to export data via CcnfRegDb command

MDBAC-220=[MDBAC-220] CcnfRegDb command not found

MDBAC-221=[MDBAC-221] failed to import data via CcnfRegDb command

MDBAC-222=[MDBAC-222] policy data not found

MDBAC-223=[MDBAC-223] Security profile rejected. Security profile of this type is disabled

MDBAC-224=[MDBAC-224] Software definition rejected. Software definition of this type is disabled

MDBAC-225=[MDBAC-225] Warning: not all matched objects are exported

MDBAC-227=[MDBAC-227] Save or Update of an database object(s) failed

MDBAC-228=[MDBAC-228] failed to read database object(s)

MDBAC-229=[MDBAC-229] creation of database user failed

Appendix A: cadsmcmd Messages

This section contains the following topics:

Return Codes

If a problem is detected by cadsmcmd, a message describing the problem is printed on standard output. This message is headed "SDCMDxxxxxx>: " or "SDCMD<Axxxxxx>: " .

- An "A" means that SD API reported the problem and the following number is the API reported error code.
- "CMD" indicates that the problem was detected and the following number is the cadsmcmd error code.

  In case of a CMD code, this code is also the return code of the cadsmcmd.

  All API codes are mapped to the cadsmcmd return code cadsmcmd_API_ERROR (32).

This section contains the following topics:

CMD000180 (see page 601)
CMD000181 (see page 601)
CMD000182 (see page 601)
CMD000183 (see page 602)

CMD000000

Successfully completed.

CMD000001

ATTIME format incorrect.

CMD000002

TIMEOUT value not numeric.

CMD000003

Invalid value assigned to a keyword.

CMD000004

Syntax error.

CMD000005

Could not open procedure file.

CMD000006

The command is allowed on the enterprise only.

CMD000007

The command is allowed on domain manager only.

CMD000008

Reserved.

CMD000009

No parameter for batch specified. The keyword batch was coded, but no batch file.

CMD000010

No parameter for local specified. The keyword local was coded, but no domain manager name.

CMD000011

Could not open batch file.

CMD000012

Error in procedure file detected. Details are listed on standard output.

CMD000013

Parameter too long.

CMD000014

Too many areas/domains, computers/computer groups or procedures.

CMD000015

Cannot combine external procedure and Software Detection script.

CMD000016

SENDTIME format is incorrect. (YYYY-MM-DD HH:MM)

CMD000017

HALTTIME format is incorrect. (YYYY-MM-DD HH:MM)

CMD000018

Filter: quotation marks not balanced.

CMD000019

Filter: parentheses not balanced.

CMD000020

Filter: logical operator expected, but not found.

CMD000021

Filter: comparative operator expected, but not found.

CMD000022

Filter: illegal comparative operator found.

CMD000023

Filter: a non-numeric value was assigned to a numeric attribute.

CMD000024

Filter: a numeric value is too large.

CMD000025

The specified system is not a scalability server.

CMD000026

No userl ID entered, login refused.

CMD000027

ECHO/UNECHO console failed. The system cannot switch between console's echo and unecho mode. Therefore, it is not possible to prompt for a password and to enter it without echo or to reverse the mode.

CMD000028

Application is running in batch. Login dialog is impossible.

CMD000029

Unknown error code received.

CMD000030

cadsmcmd abnormally ended due to an error reported before.

CMD000031

Invalid period specification.

CMD000032

API error reported. API errors have been reported before on standard output. cadsmcmd returns this code.

CMD000033

Out of memory.

CMD000034

Internal error.

CMD000035

Unknown parameter "...".

CMD000036

Invalid task.

CMD000037

Order not found.

CMD000038

The argument ... is missing.

CMD000039

The argument ...=... is invalid.

CMD000040

Either the argument ... or ... should only be coded.

CMD000041

Unknown command ...

CMD000042

DELIVERYTIME format incorrect.

CMD000043

No recoverable jobs found.

CMD000044

No recoverable jobs for the computers found.

CMD000045

Renew not allowed.

CMD000046

Wrong state.

CMD000047

Specified software template was not found.

CMD000048

Specified job was not found.

CMD000049

Either the argument ... is missing or ....

CMD000050

Either the argument ... is missing or .... or ....

CMD000051

Internal error: Root of computer groups not found.

CMD000052

Coded action not supported by domain manager.

CMD000053

Coded action not supported by enterprise manager.

CMD000054

The super group is not a software group.

CMD000055

More than one name parameter is coded.

CMD000056

Invalid syntax of itemList.

CMD000057

The group is not a procedure group.

CMD000058

The group is not a catalog group.

CMD000059

The Jobcontainer is sealed.

CMD000060

The group is a catalog group.

CMD000061

Quote error in line %d.

CMD000062

Line %d confusing.

CMD000063

Invalid type of group.

CMD000064

Software templates and computer groups of other types have been mixed.

CMD000065

Software template already sealed.

CMD000066

Insufficient permissions.

CMD000067

The job container has insufficient permissions.

CMD000068

The job container name is not unique.

CMD000069

The distribution is in another state other than "" BUILDING"" .

CMD000070

This group is not a query group.

CMD000071

This group is not associated with a software policy.

CMD000072

This group is sealed.

CMD000073

This group is not sealed.

CMD000074

This group is a query group.

CMD000075

Missing read permission.

CMD000076

Missing write permission.

CMD000077

The object cannot be removed.

CMD000078

Errors have been reported during batch processing.

CMD000079

Duplicate entry "<entry identifier>" in "<path #1>" and "<path #2>". The latter is ignored.

CMD000080

item "<item name>" of version "<item version>" could not be found among the distributed items of area "<area name>".

CMD000081

Object not found.

CMD000082

Incorrect format of date/time (YYYY-MM-DD hh:mm).

CMD000083

In the current state the modification by "<method>" is not allowed.

CMD000084

Parameter "<parameter name>" invalid in this context.

CMD000085

Parameter position is out of range.

CMD000086

No jobs to renew found.

CMD000087

"<string>" is not a keyword.

CMD000088

Target not found.

CMD000089

Batch file empty.

CMD000090

Renew is not allowed for this container.

CMD000091

The group is empty.

CMD000092

Invalid computer name.

CMD000093

Too many arguments.

CMD000094

Reporting BMS error.

Further details in BMS Errors. (see definition on page 602)

CMD000095

Reporting BMS database access error.

Further details in CSM Errors. (see definition on page 606)

CMD000096

BMSCAPI request failed.

CMD000097

List is empty.

CMD000098

End of list (internal code).

CMD000099

MAC address refers a system that is already BMS managed.

CMD000100

Product is not installed.

CMD000101

The computer is already BMS managed.

CMD000102

The computer has not been registered from the network yet.

CMD000103

There is no matching PXE computer entry in OSIM.

CMD000104

The object properties or status do not allow this action

CMD000105

Query execution error "<error>".

CMD000106

Access to query list failed. Caught error code: "<code>".

CMD000107

CO API Request failed: "<code>" - "<text>".

CMD000108

CO API indicates NULL value in database field.

CMD000109

Session establishment failed.

CMD000110

CO API Request failed: element "<number>": "<code>" - "<text>"

CMD000111

Specified boot server not found.

CMD000112

Computer user specified is of invalid format.

CMD000113

Computer not unique.

CMD000114

User not unique.

CMD000115

Software Templates not found.

CMD000116

Query object not found.

CMD000117

Computer group does not exist.

CMD000118

Warning: Function is obsolete.

CMD000119

Policy "<name>" does not exist

CMD000120

Policy: "<name>" type "<type>" does not match target type "<type>"

CMD000121

Operation not allowed: configuration "<uuid>" is not in state planned

CMD000122

Operation not allowed: configuration "<uuid>" is neither in planned nor in error state

CMD000123

Operation not allowed: configuration "<uuid>" is in planned, or in error, or in canceling or in active state

CMD000124

Operation not allowed: policy "<name>" is not in UNSEALED state

CMD000125

Value of the parameter is invalid for the type of the parameter

CMD000126

USD query creation error: code = "<code>" - text = "<text>".

CMD000127

USD QueryInit error: code = "<code>".

CMD000128

Configuration identifier "<uuid>" not found.

CMD000129

Error in procedure file detected.

CMD000130

No parameters specified, command not processed.

CMD000131

Scalability server group "<name>" not found.

CMD000132

Filter evaluation failed: <filter>

CMD000133

Scalability server "<name>" not found.

CMD000134

Group "<name>" is no server group.

CMD000135

Group "<name>" is no computer or template group.

CMD000136

Parameter information not found.

CMD000137

Parameter "<name>" does not exist.

CMD000138

Section "<name>" does not exist.

CMD000139

Section information not found.

CMD000140

Policy information not found.

CMD000141

Configuration information not found.

CMD000142

Error object information not found.

CMD000143

At least one of the multiple operations failed.

CMD000144

Operation failed.

CMD000145

CCNF API Request failed: "<code>" - "<text>"

CMD000146

Operation not allowed for Default Computer Policy.

CMD000147

Engine name "<name" not found.

CMD000148

Engine uuid "<uuid"> not found.

CMD000149

The target has already registered. The operation is not allowed.

CMD000150

MAC address missing. System not OSIM enabled.

CMD000151

The following arguments are conflicting: <arg1> and <arg2>

CMD000152

Invalid order name

CMD000153

Software policy already linked.

CMD000154

Software policy is not linked to specified computer group.

CMD000155

Computer group not found.

CMD000156

The system is an ADS device

CMD000157

The software policy is not sealed.

CMD000158

No addressable targets left.

CMD000159

Specified MAC address does not match.

CMD000160

Computer/user not found.

CMD000161

Not all jobs were removed from software policy.

CMD000162

No install procedure found.

CMD000163

No default install procedure found.

CMD000164

Software policy not supported for legacy manager "<name>".

CMD000165

Warning: software policy should be sealed but no target and/or no software are assigned.

CMD000166

Software policy cannot be sealed - no target group assigned.

CMD000167

Query "<query name>" not found.

CMD000168

Query "<query name>" is not unique.

CMD000169

Domain "<domain name>" not found.

CMD000170

Query "<query name>" for domain "<domain name>" not found.

CMD000171

Manager "<manager name>" not found.

CMD000172

Manager "<manager name>" is not unique.

CMD000173

AMAPI Error - <AMAPI Error Code> - <AMAPI error message>

CMD000174

No action available as AM is not supported by the addressed manager.

CMD000175

Class name or class id "<class name>" does not exist.

CMD000176

Security area "<security area>" not found.

CMD000177

Security object "<security object>" not found.

CMD000178

Folder "<folder name>" not found.

CMD000179

Folder id "<folder id>" not found.

CMD000180

For class id "SoftwareJob" the argument "foldername=" is missing.

CMD000181

For class id "SoftwareJob" the argument "folderclassid=" is missing.

CMD000182

Folder "__SD_Software Templates" not found.

CMD000183

Domain name "<domainName>" invalid.

BMS Errors

This section contains the following topics:

BMS 000000

Method successfully completed.

BMS 000001

Memory shortage.

BMS 000002

Missing server name.

BMS 000003

Connection to a server is still pending.

BMS 000004

No session to server established yet.

BMS 000005

This method is not implemented yet.

BMS 000006

The specified MAC address does not match the MAC address of the specified system.

BMS 000007

Internal error.

BMS 000008

The boot server is still busy with a scheduled configuration for this computer.

BMS 000009

Modification of boot status failed.

BMS 000010

The computer could not be located in database.

BMS 000010

Invalid parameter.

BMS 000012

The computer is already named.

BMS 000013

Property not found.

BMS 000014

The boot status does not suit for this method.

BMS 000015

No planned configuration available.

BMS 000016

There is already an active configuration.

BMS 000017

Invalid date/time specification.

BMS 000018

Image is not available, detected only

BMS 000019

No scheduled configuration available.

BMS 000020

Invalid configuration status.

BMS 000021

Invalid configuration specified.

BMS 000022

No current configuration available.

BMS 000023

The MAC address already exists.

BMS 000024

The computer name is not unique.

BMS 000025

Image not found.

BMS 000026

Parameter not found.

BMS 000027

For modification of operating system image use the method modifyOS.

BMS 000028

This parameter is mandatory and has no default.

BMS 000029

userParam and userValue should not be coded together.

BMS 000030

User parameter not found.

BMS 000031

The parameter value does not meet the specifications of the parameter.

BMS 000032

The MAC address is already in use by another BMS managed computer.

BMS 000033

The value is not a member of the list.

BMS 000034

The target computer is OSIM-unmanaged and has no configurations assigned.

BMS 000100

The boot server could not be located at database.

CSM Errors

This section contains the following topics:

CSM 000000

Error reported by CSM method.

CSM 000001

Connection not initialized.

CSM 000002

Function already done.

CSM 000003

Function argument invalid.

CSM 000004

Memory allocation error.

CSM 000005

Local error sending message.

CSM 000006

No reply from server.

CSM 000007

Invalid reply from server.

CSM 000008

No (more) data available.

CSM 000009

Request failed.

CSM 000010

Insufficient access permissions.

Appendix B: Supported MDB Object Types

The following table includes the MDB objects that MDB Admin Console supports:

MDB object type	Title	Notes
BoAreaDef	Area Definitions	On import, Area Definitions are not linked to Security Profiles. This means that the relationship between the Area Definitions and Security Profiles is not imported into the target MDB.
BoSecurityProfile	Security Profiles	Export includes the corresponding security class permissions for each security profile
BoQueryDef	Queries	Export includes the corresponding object permissions with respect to the security profiles. In case the security profiles exist in the target MDB, the import applies these permissions to the imported objects. The reference of the Query to its parent Query folder is maintained in the target database with the import. This requires that the parent Query folder already exist in the target database. You must manually create the Query folder. The import operation does not create this folder.
BoGroupDef	Asset Groups	Export includes the corresponding object permissions and group member permissions with respect to the security profiles. In case the security profiles exist in the target MDB, the import applies these permissions to the imported objects. In case of a dynamic group, the relationship to the query is maintained in the target database with the import. However, this requires that a query with the same name already exists in the target database. The reference of the Group to its parent Groups is maintained in the target database with the import. This requires that the parent Groups already exist in the target database.

MDB object type	Title	Notes
BoPolicyQuery Based	Query-based Policies	Export includes the corresponding object permissions with respect to the security profiles. In case the security profiles do exist in the target MDB, the import applies these permissions to the imported objects. The relationship of the Query-based Policy to the query is maintained in the target database with the import. However, this requires that a query with the same name already exists in the target database. The relationship of the Query-based Policy to an Action is not maintained by the export/import. Subfolders of the Query-based Policy folder are not maintained.
BoPolicyEvent Based	Event-based Policies	Export includes the corresponding object permissions with respect to the security profiles. In case the security profiles do exist in the target MDB, the import applies these permissions to the imported objects. The relationship of the Event-based Policy to an Action is not maintained by the export/import. Subfolders of the Event-based Policy folder are not maintained.
BoPolicySoftware	Software-based Policies	Export includes the corresponding object permissions with respect to the security profiles. In case the security profiles exist in the target MDB, the import applies these permissions to the imported objects. The relationship of the Software-based Policy to the Software Delivery Jobs is maintained in the target database with the import. However, this requires that the respective Software Delivery packages and procedures already exists in the target database. Software Delivery Library packages referenced by the Software-based Policies of the source MDB should be made available in the target MDB prior to the import. Software Delivery Library packages can be exported and imported using the corresponding methods which the EGC GUI provides or using the CADSMCMD command line interface. The reference of the Software-based Policy to Asset Groups is maintained in the target database with the import. However, this requires that the respective Asset Groups already exist in the target database.

MDB object type	Title	Notes
BoSoftware Group	Software Groups	Export includes the corresponding object permissions with respect to the security profiles. In case the security profiles exist in the target MDB, the import applies these permissions to the imported objects. The reference of the Software Group to its parent Groups is maintained in the target database with the import. This requires that the parent Groups already exist in the target database.
BoProcedure Group	Procedure Groups	Export includes the corresponding object permissions with respect to the security profiles. In case the security profiles do exist in the target MDB, the import applies these permissions to the imported objects. The reference of the Procedure Group to its parent Groups is maintained in the target database with the import. This requires that the parent Groups already exist in the target database.
BoSoftwareDef	Software Definitions	Export includes the corresponding object permissions with respect to the security profiles. In case the security profiles exist in the target MDB, the import applies these permissions to the imported objects. Export includes the Software Definitions but not the related data like software signatures, manufacturer information, and categories. You must use the Content Utility, which is available with the CA ITCM releases r12.5, r12.5SP1, r12.5SP1 C1 and r12.8 in combination with the MDB Admin Console to export and import all of this data. **Note:** Only user-defined Software Definitions are supported with the export operation. You must first use the Content Utility to export data from the source MDB and import it into the target MDB. Then use the MDB Admin Console for the Software Definitions. The MDB Admin Console export includes the corresponding object permissions with respect to the security profiles. In case the security profiles do exist in the target MDB, the import applies these permissions to the imported objects. **Note:** Software Definitions can only be exported and imported between MDB databases of the same CA ITCM relase. For example, you can export data from an r12.5 MDB and import this data only into an r12.5 MDB.

MDB object type	Title	Notes
BoPolicy Computer	Configuration Policies	For the export of configuration policies, you must install the MDB Admin Console on the DSM manager machine where you plan to run the export. This is because MDB Admin Console uses the functionality which is available on the manager. Export includes the corresponding object permissions with respect to the security profiles. In case the security profiles exist in the target MDB, the import applies these permissions to the imported objects.

Appendix C: Rules for MDB Object Comparison

The following table includes the rules that the compare method uses to compare two MDB objects:

MDB object type	Title	Attributes being compared
BoAreaDef	Area Definitions	Security Area Name
BoSecurityProfile	Security Profiles	■ Security Profile Name ■ Security Profile Type
BoQueryDef	Queries	■ Query Name ■ Query Type
BoGroupDef	Asset Groups	■ Group Name ■ Members inherit permissions ■ Group Type
BoPolicyQuery Based	Query-based Policies	■ Policy Name ■ Policy Description ■ Policy Severity
BoPolicyEvent Based	Event-based Policies	■ Policy Name ■ Policy Area ■ Policy Severity
BoPolicySoftware	Software-based Policies	■ Software Policy Name ■ Priority ■ Type
BoSoftware Group	Software Groups	■ Software Group Name ■ Type ■ Security Group
BoProcedure Group	Procedure Groups	■ Procedure Group Name ■ Type ■ Security Group

MDB object type	Title	Attributes being compared
BoSoftwareDef	Software Definitions	■ Name ■ Language ■ SW version label ■ SW version number ■ Software type id ■ Source type id ■ Lang code ■ Class id
BoPolicy Computer	Configuration Policies	Policy Name **Note:** The parameters and parameter values of the policy are not subject to the comparison.

Index

%

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V