WebSphere® Application Server V4.0 for z/OS and OS/390

System Management Scripting API

SA22-7839-01

WebSphere® Application Server V4.0 for z/OS and OS/390

System Management Scripting API

SA22-7839-01

Note

Before using this information and the product it supports, be sure to read the general information under "Appendix. Notices" on page 237.

Second Edition (June 2001)

This is a major revision of SA22–7839–00

This edition applies to WebSphere Application Server V4.0 for z/OS and OS/390 (5655-F31), and to all subsequent releases and modifications until otherwise indicated in new editions.

The most current versions of the WebSphere Application Server V4.0 for z/OS and OS/390 publications are at this Web site: `http://www.ibm.com/software/webservers/appserv/`

Contents

Figures

About this book

This book introduces the IBM WebSphere Application Server V4.0 for z/OS and OS/390 Systems Management Scripting API product. It describes the major functions and features of the product and gives a short reference about how to upgrade a running system for the SM Scripting API. This book is not meant to replace documentation for the Systems Management - End User Interface (SM-EUI). It only describes the functionality of the SM Scripting API.

Note: The full product name is ″WebSphere Application Server V4.0 for z/OS and OS/390,″ hereafter referred to in this text as ″WebSphere for z/OS″ or the ″Application Server.″

How this book is organized

The following is an overview of the chapter order and contents.

- "Chapter 1. Introduction" on page 1 introduces the SM Scripting API.
- "Chapter 2. How to install the SM Scripting API" on page 3 describes how to install (upgrade to) the SM Scripting API on a system that is already running.
- "Chapter 3. CB390CMD" on page 5 describes specifics of the command processor CB390CMD.
- "Chapter 4. CB390CFG" on page 17 describes specifics of CB390CFG.
- "Chapter 5. XMLGEN" on page 169 describes specifics of the REXX script XMLGEN.
- "Chapter 6. XMLPARSE" on page 173 describes specifics of the REXX script XMLPARSE.
- "Chapter 7. XMLFIND" on page 177 describes specifics of the REXX script XMLFIND.
- "Chapter 8. XMLEXTRACT" on page 181 describes specifics of the REXX script XMLEXTRACT.
- "Chapter 9. Default XML files" on page 185 lists the default values of the SM-EUI which can be modified by the user.
- "Chapter 10. Sample REXX scripts" on page 227 provides examples for various actions you can take with REXX scripts.
- "Appendix. Notices" on page 237 provides general information about this book.

Where to find related information

This is a list of books that are in the WebSphere for z/OS library. They can be found at the following Web site:

`http://www.ibm.com/software/webservers/appserv/`

- *WebSphere Application Server V4.0 for z/OS and OS/390: Program Directory*, GI10-0680, describes the elements of and the installation instructions for WebSphere for z/OS.
- *WebSphere Application Server V4.0 for z/OS and OS/390: License Information*, LA22-7855, describes the license information for WebSphere for z/OS.
- *WebSphere Application Server V4.0 for z/OS and OS/390: Installation and Customization*, GA22-7834, describes the planning, installation, and customization tasks and guidelines for WebSphere for z/OS.
- *WebSphere Application Server V4.0 for z/OS and OS/390: Messages and Diagnosis*, GA22-7837, provides diagnosis information and describes messages and codes associated with WebSphere for z/OS.
- *WebSphere Application Server V4.0 for z/OS and OS/390: Operations and Administration*, SA22-7835, describes system operations and administration tasks.
- *WebSphere Application Server V4.0 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836, describes how to develop, assemble, and install J2EE applications in a WebSphere for z/OS J2EE server. It also includes information about migrating applications from previous releases of WebSphere Application Server for OS/390, or from other WebSphere family platforms.
- *WebSphere Application Server V4.0 for z/OS and OS/390: Assembling CORBA Applications*, SA22-7848, describes how to develop, assemble, and deploy CORBA applications in a WebSphere for z/OS (MOFW) server.
- *WebSphere Application Server V4.0 for z/OS and OS/390: System Management User Interface*, SA22-7838, describes the system administration and operations tasks as provided in the Systems Management User Interface.
- *WebSphere Application Server V4.0 for z/OS and OS/390: System Management Scripting API*, SA22-7839, describes the functionality of the WebSphere for z/OS Systems Management Scripting API product.

You might also need to refer to information about other z/OS or OS/390 elements and products. All of this information is available through links at the following Internet locations:

```
http://www.ibm.com/servers/eserver/zseries/zos/
http://www.ibm.com/servers/s390/os390/
```

Here are some books that you might find particularly helpful:

- *Getting Started with WebSphere Application Server*, SC09-4581, provides an overview of WebSphere for z/OS and describes requirements for setting up the environment.
- *Building Business Solutions with WebSphere*, SC09-4432

How to send your comments

Your feedback is important in helping to provide the most accurate and highest quality information. You can e-mail your comments to:

```
wasdoc@us.ibm.com
```

or fax them to 919-254-0206.

Be sure to include the document name and number, the WebSphere Application Server version, and, if applicable, the specific page, table, or figure number on which you are commenting.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Summary of changes

Summary of changes
for SA22–7839–01
WebSphere for z/OS
as updated, June 2001,
service level W400018

This book contains information previously presented in SA22–7839–00, which supports WebSphere for z/OS. The following is a summary of changes to this information:

The following APARs required changes to this book:

- APAR PQ48859 (PTF UQ54362, service level W400012)
- APAR PQ49215 (PTF UQ54755, service level W400014)

Changes have been made to the following topics:

- “Steps for setting up the client environment” on page 3
- “Chapter 4. CB390CFG” on page 17
- “Chapter 5. XMLGEN” on page 169
- “inputcreateserver.xml” on page 192
- “inputcreatej2eeserver.xml” on page 201

Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Chapter 1. Introduction

This chapter introduces the SM Scripting API.

The SM Scripting API

The SM Scripting API is an additional feature of the WebSphere for z/OS Systems Management. The functionality that is provided by the SM Scripting API is exactly the same as the SM-EUI, which is already part of WebSphere for z/OS. To use the SM Scripting API, the user needs to know how to write a REXX script, because REXX is, for this moment, the only script language that is supported by the SM Scripting API.

A typical SM Scripting API script consists of these three parts:

- One part to generate the input file
- One part to call the function
- One part to process the result

For these three parts there are already functions written in REXX to make life easier. These functions are:

- `CB390CMD(...)` for calling an operations function
- `CB390CFG(...)` for calling an administrations function
- `XMLGEN(...)` for generating the input file
- `XMLPARSE(...)` for parsing the result
- `XMLFIND(...)` for finding a special attribute or value
- `XMLEXTRACT(...)` for extracting a known attribute

To use one of the administration functions, the default xml files must be present. In these default xml files, there are listed all valid attributes for each action. The style of the document is given by the Document Type Definition (DTD).

Chapter 2. How to install the SM Scripting API

The SM Scripting API is now a fully-intergrated part of WebSphere for z/OS, so it comes pre-installed. Therefore, the only purpose of this chapter is to help you set up the client environment.

Note: The product that is referred to in this text as the ″Administration application″ may be seen elsewhere with one of the following informal names:

- SM-EUI
- SM-GUI
- GUI

Steps for setting up the client environment

Important: These steps must be performed each time you login to OMVS.

Perform these steps to set up the client environment:

1. Open OMVS.
2. To enable Java clients on z/OS or OS/390, add the following values to the environment variables LIBPATH and CLASSPATH:
   ```
   export LIBPATH=$LIBPATH:/lib:/usr/lpp/java/IBM/J1.3/bin:/usr/lpp/java/IBM
   /J1.3/bin/classic:/usr/lpp/WebSphere/lib
   ```
   (The above command should be all on one line.)
   ```
   export CLASSPATH=$CLASSPATH:path/ws390crt.jar
   ```
   where the default *path* for these files is `/usr/lpp/WebSphere/lib`.

   To learn more about the environment variables LIBPATH, CLASSPATH, and PATH, refer to the chapter ″Application development and client environments″ in *WebSphere Application Server V4.0 for z/OS and OS/390: Installation and Customization*, GA22-7834.
3. Add the new JAR (Java Archive) files to the existing CLASSPATH. This is done by typing the following commands (these are type sensitive):
   ```
   export CLASSPATH=$CLASSPATH:/usr/lpp/WebSphere/lib/xerces.jar
   export CLASSPATH=$CLASSPATH:/usr/lpp/WebSphere/lib/ws390sms.jar
   ```
4. Add a new path to the PATH environment variable by typing the following command (this is case sensitive):
   ```
   export PATH=$PATH:/usr/lpp/WebSphere/bin
   ```

5. Add a new environment variable by typing the following command. This is case sensitive:

```
export DEFAULT_CLIENT_XML_PATH=/usr/lpp/WebSphere/samples/smapi
```

6. You can make sure that the settings have taken hold by typing the following commands:

```
echo $CLASSPATH
echo $DEFAULT_CLIENT_XML_PATH
echo $PATH
```

   In all cases, the settings you have made before must be added to the end of each string.

Make sure that the userid you use for OMVS is registered as a WebSphere for z/OS Systems Management Administrator. CBADMIN works well in this case. If you don’t use CBADMIN, you need to define another userid to be a Systems Management Administrator. See the section "Adding a new administrator for the Administration application" in the *WebSphere Application Server V4.0 for z/OS and OS/390: Installation and Customization*, GA22-7834 for more information on how to define a new administrator via the Administration application (SM EUI)

Notes:

1. The scripts cannot be run from another userid that has done a "switch user" to become CBADMIN.
2. IBM provides a sample file called *client_environment* to which you can refer for more information. This file can be found in the directory /usr/lpp/WebSphere/samples/smapi.

That’s it. Now you can use the SM Scripiting API. Have fun writing your REXX scripts!

Chapter 3. CB390CMD

CB390CMD is a command processor whose purpose is to control servers or server instances in the active configuration. The SM Scripting API provides the same functionality as the SM Operations EUI.

Syntax

```
►►──rc = CB390CMD──("── -action──’ ──┬─start─────────┬──’ ─────────────────────►
                                     ├─stop──────────┤
                                     ├─cancel────────┤
                                     ├─cancelrestart─┤
                                     └─list──────────┘

►─ -server──’servername’ ── -serverinstance──’serverinstancename’ ────────────►

►─ -output──’outputfilename’ ──") ────────────────────────────────────────────►◄
```

Syntax details

rc Return code from performed operation. This signals if the operation ended successfully (rc = 0) or if an error occurred (rc = 4).

-action
Describes the function that should be performed. The following functions are implemented in the CB390CMD API:

start	Causes the server or server instance to be started
stop	Causes the server or server instance to be stopped
cancel	Causes the server or server instance to be canceled
cancelrestart	Causes the server or server instance to be canceled and restarted
list	Causes the server or server instance to be listed

-server
Name of the server that is being operated on.

-serverinstance
Name of the server instance that is being operated on.

-output
: Temporary output file that stores the results of the action. The output file contains further information. All server instances are listed there, including the belonging server and server instance state. In the descriptions of the functions, there are examples of the output. The functions XMLPARSE ("Chapter 6. XMLPARSE" on page 173), XMLFIND ("Chapter 7. XMLFIND" on page 177) and XMLEXTRACT ("Chapter 8. XMLEXTRACT" on page 181) are used to work with the output file.

Action "start"

This action causes the server or server instance to be started.

Syntax

```
►►──rc = CB390CMD──("── -action──'start' ── -server──'servername' ──────────►

►──┬──────────────────────────────────────────────┬──────────────────────────►
   └─ -serverinstance──'serverinstancename' ──────┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────►◄
```

Syntax details

rc
: The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "4" an error has occurred while processing the action.

servername
: This parameter specifies the name of the server where the function start should be operated on. If the server name equals "*" then the specified server instance will be started in all servers it pertains to. Example: If the server name is "*" and the server instance name is "*" then all serverinstances in all server will be started.

serverinstancename
: This parameter specifies the name of the server instance where the function start should operate on. If the server instance name equals "*" or is not present then all server instances will be started in the specified server.

outputfilename
: This parameter specifies the file name where the output data of the function start should be written to. To work with the output file there are several functions like XMLPARSE ("Chapter 6.

XMLPARSE” on page 173), XMLFIND (“Chapter 7. XMLFIND” on page 177) and XMLEXTRACT (“Chapter 8. XMLEXTRACT” on page 181).

Example

Here is an example REXX script that starts all server instances in server ″BBOASR1″. After that the REXX script `XMLPARSE` will be used to list the output file on the screen.

```
/* REXX function */
call syscalls 'ON'
signal on error

rc = 0

rc = CB390CMD("-action 'start' -servername 'BBOASR1'
               -serverinstancename '*' -output 'FCT33'")
if (rc == 4) then do
  say "FCT Test #33 failed"
  exit
end

rc = XMLPARSE("FCT33"  "ALL")
if (rc == 4) then do
  say "FCT Test #33 failed while XMLPARSE"
  exit
end
say "FCT Test #33 completed"
exit

error:
say "Error in FCT Test #33" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
servername.1 BBOASR1
serverinstancename.1 BBOASR1A
serverinstancestatus.1 Active
servername.2 BBOASR1
serverinstancename.2 BBOASR1B
serverinstancestatus.2 Active
status 0
message.1 OK
count 2
```

Action “stop”

This action causes the server or server instance to be stopped. After that the REXX script `XMLPARSE` will be used to list the output file on the screen.

Syntax

```
►►──rc = CB390CMD──("── -action──'stop' ── -server──'servername' ──────────────►

►──┬─────────────────────────────────────────────┬──────────────────────────────►
   └─ -serverinstance──'serverinstancename' ──┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is ″0″ if everything ended correctly. If the return code (rc) is ″4″ an error has occurred while processing the action.

servername
This parameter specifies the name of the server where the function stop should be operated on. If the server name equals ″*″ then the specified server instance will be stopped in all servers it pertains to. Example: If the server name is ″*″ and the server instance name is ″*″ then all serverinstances in all server will be stopped.

serverinstancename
This parameter specifies the name of the server instance where the function stop should operate on. If the server instance name equals ″*″ or is not present then all server instances will be stopped in the specified server.

outputfilename
This parameter specifies the file name where the output data of the func-tion stopped should be written to. To work with the output file there are several functions like XMLPARSE (“Chapter 6. XMLPARSE” on page 173), XMLFIND (“Chapter 7. XMLFIND” on page 177) and XMLEXTRACT (“Chapter 8. XMLEXTRACT” on page 181).

Example
Here is an example REXX script that stops all server instances in server ″BBOASR1″. After that the REXX script XMLPARSE will be used to list the output file on the screen.

```
/* REXX function */
call syscalls 'ON'
signal on error

rc = 0

rc = CB390CMD("-action 'stop' -servername 'BBOASR1'
               -serverinstancename '*' -output 'FCT34'")
if (rc == 4) then do
  say "FCT Test #34 failed"
  exit
end

rc = XMLPARSE("FCT34"  "ALL")
if (rc == 4) then do
  say "FCT Test #34 failed while XMLPARSE"
  exit
end
say "FCT Test #34 completed"
exit

error:
say "Error in FCT Test #34" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
servername.1 BBOASR1
serverinstancename.1 BBOASR1A
serverinstancestatus.1 Stopped
servername.2 BBOASR1
serverinstancename.2 BBOASR1B
serverinstancestatus.2 Stopped
status 0
message.1 OK
count 2
```

Action “cancel”

This action causes the server or server instance to be canceled.

Syntax

```
►►—rc = CB390CMD—("— -action—'cancel' — -server—'servername' ——————————►

►——————————————————————————————————————————————————————————————————————►
  └ -serverinstance—'serverinstancename' ┘

►— -output—'outputfilename' —")————————————————————————————————————————►◄
```

Syntax details

rc The return code (rc) is ″0″ if everything ended correctly. If the return code (rc) is ″4″ an error has occurred while processing the action.

servername
This parameter specifies the name of the server where the function cancel should be operated on. If the server name equals ″*″ then the specified server instance will be canceled in all servers it pertains to. Example: If the server name is ″*″ and the server instance name is ″*″ then all serverinstances in all server will be canceled.

serverinstancename
This parameter specifies the name of the server instance where the function cancel should operate on. If the server instance name equals ″*″ or is not present then all server instances will be canceled in the specified server.

outputfilename
This parameter specifies the file name where the output data of the function cancel should be written to. To work with the output file there are several functions like XMLPARSE (“Chapter 6. XMLPARSE” on page 173), XMLFIND (“Chapter 7. XMLFIND” on page 177) and XMLEXTRACT (“Chapter 8. XMLEXTRACT” on page 181).

Example

Here is an example REXX script that cancels all server instances in server ″BBOASR1″. After that the REXX script `XMLPARSE` will be used to list the output file on the screen.

```
/* REXX function */
call syscalls 'ON'
signal on error

rc = 0

rc = CB390CMD("-action 'cancel' -servername 'BBOASR1'
               -serverinstancename '*' -output 'FCT35'")
if (rc == 4) then do
  say "FCT Test #35 failed"
  exit
end

rc = XMLPARSE("FCT35"  "ALL")
if (rc == 4) then do
  say "FCT Test #35 failed while XMLPARSE"
  exit
end
say "FCT Test #35 completed"
```

```
exit

error:
say "Error in FCT Test #35" rc "at line" sigl
say sourceline(sigl)
exit
```

The output of this function is very critical, because the server instance the function performs on may not have been canceled immediately. So the state that is shown for the server instance here may not be the correct one.

The output file may look like this:

```
servername.1 BBOASR1
serverinstancename.1 BBOASR1A
serverinstancestatus.1 Stopped
servername.2 BBOASR1
serverinstancename.2 BBOASR1B
serverinstancestatus.2 Stopped
status 0
message.1 OK
count 2
```

Action “cancelrestart”

This action causes the server or server instance to be canceled and restarted.

Syntax

```
►►──rc = CB390CMD──("── -action──'cancelrestart' ── -server──'servername' ──────►

►─┬───────────────────────────────────────────┬──────────────────────────────────►
  └─ -serverinstance──'serverinstancename' ──┘

►─ -output──'outputfilename' ──")─────────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is ″0″ if everything ended correctly. If the return code (rc) is ″4″ an error has occurred while processing the action.

servername
This parameter specifies the name of the server where the function cancelrestart should be operated on. If the server name equals ″*″ then the specified server instance will be canceled and restarted in all servers it pertains to. Example: If the server name

is ″*″ and the server instance name is ″*″ then all serverinstances in all server will be canceled and restarted.

serverinstancename
This parameter specifies the name of the server instance where the function cancelrestart should operate on. If the server instance name equals ″*″ or is not present then all server instances will be canceled and restarted in the specified server.

outputfilename
This parameter specifies the file name where the output data of the function cancelrestart should be written to. To work with the output file there are several functions like XMLPARSE (“Chapter 6. XMLPARSE” on page 173), XMLFIND (“Chapter 7. XMLFIND” on page 177) and XMLEXTRACT (“Chapter 8. XMLEXTRACT” on page 181).

Example

Here is an example REXX script that cancels and restarts all server instances in server ″BBOASR1″. After that the REXX script XMLPARSE will be used to list the output file on the screen.

```
/* REXX function */
call syscalls 'ON'
signal on error

rc = 0

rc = CB390CMD("-action 'cancelrestart' -servername 'BBOASR1'
              -serverinstancename '*' -output 'FCT36'")
if (rc == 4) then do
  say "FCT Test #36 failed"
  exit
end

rc = XMLPARSE("FCT36"  "ALL")
if (rc == 4) then do
  say "FCT Test #36 failed while XMLPARSE"
  exit
end
say "FCT Test #36 completed"
exit

error:
say "Error in FCT Test #36" rc "at line" sigl
say sourceline(sigl)
exit
```

The output of this function is very critical, because the server instance the function performs on may not have been canceled and restarted immediately. So the state that is shown for the server instance here may not be the correct one. The output file may look like this:

```
servername.1 BBOASR1
serverinstancename.1 BBOASR1A
serverinstancestatus.1 Stopped
servername.2 BBOASR1
serverinstancename.2 BBOASR1B
serverinstancestatus.2 Stopped
status 0
message.1 OK
count 2
```

Action “list”

This action causes the server or server instance to be listed.

Syntax

```
►►──rc = CB390CMD──("── -action──'list' ── -server──'servername' ──────────►

►──┬────────────────────────────────────────────────┬───────────────────────►
   └─ -serverinstance──'serverinstancename' ──┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "4" an error has occurred while processing the action.

servername
This parameter specifies the name of the server where the function list should be operated on. If the server name equals "*" then the specified server instance will be listed in all servers it pertains to. Example: If the server name is "*" and the server instance name is "*" then all serverinstances in all server will be listed.

serverinstancename
This parameter specifies the name of the server instance where the function list should operate on. If the server instance name equals "*" or is not present then all server instances will be listed in the specified server.

outputfilename
This parameter specifies the file name where the output data of the function list should be written to. To work with the output file there are several functions like XMLPARSE (“Chapter 6.

XMLPARSE" on page 173), XMLFIND ("Chapter 7. XMLFIND" on page 177) and XMLEXTRACT ("Chapter 8. XMLEXTRACT" on page 181).

Example

Here is an example REXX script that lists all server instances in all server. After that the REXX script XMLPARSE will be used to list the output file on the screen.

```
/* REXX function */
call syscalls 'ON'
signal on error

rc = 0

rc = CB390CMD("-action 'list' -servername '*'
               -serverinstancename '*' -output 'FCT47'")
if (rc == 4) then do
  say "FCT Test #47 failed"
  exit
end

rc = XMLPARSE("FCT47"  "ALL")
if (rc == 4) then do
  say "FCT Test #47 failed while XMLPARSE"
  exit
end
say "FCT Test #47 completed"
exit

error:
say "Error in FCT Test #47" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
servername.1 BBOASR1
serverinstancename.1 BBOASR1A
serverinstancestatus.1 Stopped
servername.2 BBOASR1
serverinstancename.2 BBOASR1B
serverinstancestatus.2 Stopped
servername.3 BBOASR2
serverinstancename.3 BBOASR2A
serverinstancestatus.3 Stopped
servername.4 BBOASR2
serverinstancename.4 BBOASR2B
serverinstancestatus.4 Stopped
servername.5 BBOASR3
serverinstancename.5 BBOASR3A
serverinstancestatus.5 Stopped
servername.6 BBOASR3
serverinstancename.6 BBOASR3B
serverinstancestatus.6 Stopped
```

```
servername.7 CBDAEMON
serverinstancename.7 DAEMON01
serverinstancestatus.7 Active
servername.8 CBINTFRP
serverinstancename.8 INTFRP01
serverinstancestatus.8 Active
servername.9 CBNAMING
serverinstancename.9 NAMING01
serverinstancestatus.9 Active
servername.10 CBSYSMGT
serverinstancename.10 SYSMGT01
serverinstancestatus.10 Active
status 0
message.1 OK
count 10
```

Chapter 4. CB390CFG

CB390CFG is used to configure z/OS or OS/390 servers and applications provides the same functionality as the SM Administration EUI.

Syntax

```
►►──rc = CB390CFG──("── -action──'actionname' ──────────────────────────────►

►── -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬───────────►
                                     └─ -input──'inputfilename' ─┘

►── -output──'outputfilename' ──")──────────────────────────────────────────►◄
```

Syntax Details

rc Return code from performed operation. This signals if the operation ended successfully (rc = 0) or if an error occurred (rc = 4).

-action
Describes the function that should be performed. The actions are described in the the following sections.

-xmlinput
Name of the default xml file. This is a xml file that contains the attributes that are required for the operation. The document type definition (DTD) is coded here. All default xml files are listed in "Chapter 9. Default XML files" on page 185.

-input
Name of the temporary input xml file. This parameter is optional. If it is set then the REXX function XMLGEN should be used to merge the default xml file with the parameter specified in the input file. The function XMLGEN is descibed in the section "Chapter 5. XMLGEN" on page 169.

-output
Temporary output file that stores the result of the action. The output file contains further information. In the description of the functions there are examples of the output. To work with the output file there are several functions like XMLPARSE

("Chapter 6. XMLPARSE" on page 173), XMLFIND ("Chapter 7. XMLFIND" on page 177) and XMLEXTRACT ("Chapter 8. XMLEXTRACT" on page 181).

Conversations

These functions are for the modifications of a conversation.

Syntax

```
►►──rc = CB390CFG──("── -action──'──┬─create- ─┬──conversation──'──────────►
                                    ├─delete- ─┤
                                    ├─commit- ─┤
                                    └─list-  ──┘

►─ -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬─────────►
                                    └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")──────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "4" an error has occurred while processing the action.

-action

createconversation
Causes a new conversation to be created.

deleteconversation
Causes the conversation to be deleted.

commitconversation
Causes the conversation to be commited and activated.

listconversation
Causes the conversation to be listed.

-xmlinput
This is the default xml file. In this file all required parameters for the action which should be performed must be specified. This file is a xml file with a document type definition (DTD). The DTD only specifies the structure of the document. The user can specify default values for each parameter. These parameters can be overriden by the REXX script via the input parameter. All default xml files are listed in "Chapter 9. Default XML files" on page 185. The parameters in these files are set to the default values of the

SM Administration EUI.
The default xml file must be present in the path that is specified by the environment variable `DEFAULT_CLIENT_XML_PATH` or the user has to specify its path.

Example: `-xmlinput 'inputcreateconversation.xml'` specifies the default input xml file in the `DEFAULT_CLIENT_XML_PATH`. But `-xmlinput './inputcreateconversation.xml'` specifies the file in the current directory.

The user can modify the default path for the default xml files by setting the environment variable `DEFAULT_CLIENT_XML_PATH` to another existing path. Make sure that the path exists and that the default xml file that will be used is in this directory.

-input
This is an optional parameter for the CB390CFG API. It specifies the input file which contains the name value pairs that should override the parameters of the default xml file. To generate a xml file by using REXX variables there is a tool called XMLGEN. This tool is described in "Chapter 5. XMLGEN" on page 169.

Important: The input file will be deleted after the parameter is merged with the default xml file.

-output
The output file contains further information. It will be written into the "/tmp" directory. In the description of each conversation action there is an example output file. The general output format for a conversation action looks like this:

```
administratorname.1 AdministratorName
conversationdescription.1 ConversationDescription
conversationname.1 ConversationName
status 0|4
message.1 OK|ErrorMessage
count NumberOfListedConversations
```

Action "createconversation"

This action causes a new conversation to be created. This new conversation is a copy of the active conversation.

Syntax

```
►►──rc = CB390CFG──("── -action──'createconversation' ──────────────────────►

►── -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬──────────►
                                     └─ -input──'inputfilename' ─┘
►─ -output──'outputfilename' ──")──────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for createconversation "inputcreateconversation. xml" is listed in section "inputcreateconversation.xml" on page 185. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `'inputcreateconversation.xml`. Otherwise specify the complete location to the default xml file by setting this parameter to `'/usr/lpp/WebSphere/samples/smapi/inputcreateconversation.xml'`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this conversation action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputcreateconversation.xml" on page 185.

Parameter name	Values	Required
conversationname	Name of the conversation	x
conversationdescription	Description of the conversation	

Example

Here is an example script:

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "conversationdescription"

val. = 0
val.1 = "API Functiontest"
val.2 = "Conversation for the Function Test"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #03 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'createconversation' -xmlinput
'inputcreateconversation.xml' -input 'tempin'
-output 'FCT03'")
if (rc == 4) then do
  say "FCT Test #03 failed"
  exit
end
exit

error:
say "Error in FCT Test #03" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
conversationdescription.1 Conversation for the Function Test
conversationname.1 API Functiontest
status 0
message.1 OK
count 1
```

Action “deleteconversation”

This action causes the named conversation to be deleted.

Syntax

```
►►──rc = CB390CFG──("── -action──'deleteconversation' ──────────────────────►

►── -xmlinput──'defaultxmlfilename' ──┬──────────────────────────────┬──────►
                                      └── -input──'inputfilename' ───┘

►── -output──'outputfilename' ──") ────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for deleteconversation "inputdeleteconversation. xml" is listed in section “inputdeleteconversation.xml” on page 186. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

  If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputdeleteconversation.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputdeleteconversation.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
: This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
: This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this conversation action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN (“Chapter 5. XMLGEN” on page 169) script. The default xml file is listed in section “inputdeleteconversation.xml” on page 186.

Parameter name	Values	Required
conversationname	Name of the conversation	x

Example

Here is an example script:

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"

val. = 0
val.1 = "API Functiontest"

rc=4
i=1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #05 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'deleteconversation' -xmlinput
'inputdeleteconversation.xml' -input 'tempin' -output
'FCT05'")
if (rc == 4) then do
  say "FCT Test #05 failed"
  exit
end
exit

error:
say "Error in FCT Test #05" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
conversationdescription.1 Conversation for the Function Test
conversationname.1 API Functiontest
status 0
message.1 OK
count 1
```

Action “commitconversation”

This causes the named conversation to be committed and activated.

Syntax

```
►►──rc = CB390CFG──("── -action──'commitconversation' ──────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬─────────────────────────────┬─────────────►
                                    └─ -input──'inputfilename' ──┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for commitconversation "inputcommitconversation. xml" is listed in section “inputcommitconversation.xml” on page 186. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputcommitconversation.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputcommitconversation.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters. outputfilename This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this conversation action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputcommitconversation.xml" on page 186.

Parameter name	Values	Required
conversationname	Name of the conversation	x

Example

Here is an example script:

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"

val. = 0
val.1 = "API Functiontest"

rc=4
i=1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #04 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'commitconversation' -xmlinput
'inputcommitconversation.xml'
               -input 'tempin' -output 'FCT04'")
if (rc == 4) then do
  say "FCT Test #04 failed"
  exit
end
exit

error:
say "Error in FCT Test #04" rc "at line" sigl
say sourceline(sigl)

exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
conversationdescription.1 Conversation for the Function Test
conversationname.1 API Functiontest
status 0
message.1 OK
count 1
```

Action “listconversation”

This causes the named conversations to be listed. If the conversation name equals "*", then all conversations will be listed.

Syntax

```
►►──rc = CB390CFG──("── -action──'listconversation' ─────────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬────────────────────────────┬───►
                                    └─ -input──'inputfilename' ──┘

►─ -output──'outputfilename' ──")───────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for listconversation "inputlistconversation.xml" is listed in section “inputlistconversation.xml” on page 187. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

  If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputlistconversation.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputlistconversation.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
: This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these

new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this conversation action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputlistconversation.xml" on page 187.

Parameter name	Values	Required
conversationname	Name of the conversation "*" to list all conversations pertaining to the Administrator	x

Example

Here is an example script:

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"

val. = 0
val.1 = "API Functiontest"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #06 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'listconversation' -xmlinput
'inputlistconversation.xml' -input 'tempin' -output
'FCT06'")
if (rc == 4) then do
  say "FCT Test #06 failed"
  exit
end
```

```
exit

error:
say "Error in FCT Test #06" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
conversationdescription.1 Conversation for the Function Test
conversationname.1 API Functiontest
status 0
message.1 OK
count 1
```

Sysplex

These functions are for the modifications of a sysplex.

Syntax

```
►►──rc = CB390CFG──("── -action──'──┬─change-─┬──sysplex──'──────────────►
                                    └─list-───┘

►── -xmlinput──'defaultxmlfilename' ──┬──────────────────────────┬───────►
                                      └─ -input──'inputfilename' ─┘

►── -output──'outputfilename' ──")───────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "4" an error has occurred while processing the action.

-action

changesysplex
Causes the sysplex to be changed.

listsysplex
Causes the sysplex to be listed.

-xmlinput
This is the default xml file. In this file all required parameters for the action which should be performed must be specified. This file is a xml file with a document type definition (DTD). The DTD only specifies the structure of the document. The user can specify

default values for each parameter. These parameters can be overriden by the REXX script via the input parameter. All default xml files are listed in "Chapter 9. Default XML files" on page 185. The parameters in these files are set to the default values of the SM Administration EUI.
The default xml file must be present in the path that is specified by the environment variable `DEFAULT_CLIENT_XML_PATH` or the user has to specify its path.

Example: `-xmlinput 'inputchangesysplex.xml'` specifies the default input xml file in the `DEFAULT_CLIENT_XML_PATH`. But `-xmlinput './inputchangesysplex.xml'` specifies the file in the current directory.

The user can modify the default path for the default xml files by setting the environment variable `DEFAULT_CLIENT_XML_PATH` to another existing path. Make sure that the path exists and that the default xml file that will be used is in this directory.

-input
This is an optional parameter for the CB390CFG API. It specifies the input file which contains the name value pairs that should override the parameters of the default xml file. To generate a xml file by using REXX variables there is a tool called XMLGEN. This tool is described in "Chapter 5. XMLGEN" on page 169.

Important: The input file will be deleted after the parameter is merged with the default xml file.

-output
The output file contains further information. It will be written into the "/tmp" directory. In the description of each sysplex action there is an example output file.

Action "changesysplex"

This action causes attributes of the sysplex to be changed.

Syntax

```
►►──rc = CB390CFG──("── -action──'changesysplex' ──────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬──────────────────────────┬─────────►
                                     └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")───────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for changesysplex "inputchangesysplex.xml" is listed in section "inputchangesysplex.xml" on page 188. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `'inputchangesysplex.xml`. Otherwise specify the complete location to the default xml file by setting this parameter to `'/usr/lpp/WebSphere/samples/smapi/inputchangesysplex.xml'`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this sysplex action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputchangesysplex.xml" on page 188.

Parameter name	Values	Required
conversationname	Name of the conversation	x
sysplexname	Name of the sysplex	x
sysplexdescription	Description of the sysplex	
logstreamname	Name of the logstream	x

Example

Here is an example script:

```
/* REXX function */
/* Functiontest Test: changesysplex*/
/* Dependencies: */
/* The sysplex "PLEX1" must be added*/
/* The conversation "API Functiontest" must be added*/

call syscalls 'ON'
signal on error

say "FCT Test changesysplex"

name. = 0
name.1  = "sysplexname"
name.2  = "sysplexdescription"
name.3  = "environment"
name.4  = "conversationname"

val. = 0
val.1  = "PLEX1"
val.2  = "My new description"
val.3  = "CLASSPATH='/usr/lpp/WebSphere/jars' PATH='/usr/lpp/WebSphere/bin'
DEFAULT_CLIENT_XML_PATH='/sm/xml'"
val.4  = "API Functiontest"

rc = 4
i = 1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test changesysplex failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: changesysplex */
rc = CB390CFG("-action 'changesysplex' -xmlinput 'inputchangesysplex.xml'
-input 'tempin' -output 'changesysplex'")
if (rc == 4) then do
  say "FCT Test changesysplex failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("changesysplex"  "ALL")
if (rc == 4) then do
  say "FCT Test changesysplex failed while XMLPARSE"
  exit
end
say "FCT Test changesysplex completed"
return 0

exit
```

```
error:
say "Error in FCT Test changesysplex" rc "at line" sigl
say sourceline(sigl)
exit
```

Action “listsysplex”

This action causes attributes of the sysplex to be listed.

Syntax

```
►►──rc = CB390CFG──("── -action──'listsysplex' ─────────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ────────────────────────────────────────►
                                     └─ -input──'inputfilename' ──┘

►─ -output──'outputfilename' ──") ─────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for listsysplex "inputlistsysplex.xml" is listed in section “inputlistsysplex.xml” on page 188. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

  If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `'inputlistsysplex.xml`. Otherwise specify the complete location to the default xml file by setting this parameter to `'/usr/lpp/WebSphere/samples/smapi/inputlistsysplex.xml'`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
: This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these new specified values. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
: This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this sysplex action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputlistsysplex.xml" on page 188.

Parameter name	Values	Required
conversationname	Name of the conversation	x
sysplexname	Name of the sysplex	x

System

These functions are for the modifications of a system.

Syntax

```
►►──rc = CB390CFG──("── -action──'──┬─create- ─┬──system──'─────────────────►
                                    ├─delete- ─┤
                                    ├─change- ─┤
                                    └─list- ───┘

►─ -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬────────────►
                                    └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "4" an error has occurred while processing the action.

-action

createsystem
: Causes a new system to be created.

deletesystem
: Causes the system to be deleted.

changesystem
: Causes the system to be changed.

listsystem
: Causes the system to be listed.

-xmlinput
: This is the default xml file. In this file all required parameters for the action which should be performed must be specified. This file is a xml file with a document type definition (DTD). The DTD only specifies the structure of the document. The user can specify default values for each parameter. These parameters can be overriden by the REXX script via the input parameter. All default xml files are listed in “Chapter 9. Default XML files” on page 185. The parameters in these files are set to the default values of the SM Administration EUI.
The default xml file must be present in the path that is specified by the environment variable `DEFAULT_CLIENT_XML_PATH` or the user has to specify its path.

  Example: `-xmlinput 'inputcreatesystem.xml'` specifies the default input xml file in the `DEFAULT_CLIENT_XML_PATH`. But `-xmlinput './inputcreatesystem.xml'` specifies the file in the current directory.

  The user can modify the default path for the default xml files by setting the environment variable `DEFAULT_CLIENT_XML_PATH` to another existing path. Make sure that the path exists and that the default xml file that will be used is in this directory.

-input
: This is an optional parameter for the CB390CFG API. It specifies the input file which contains the name value pairs that should override the parameters of the default xml file. To generate a xml file by using REXX variables there is a tool called XMLGEN. This tool is described in “Chapter 5. XMLGEN” on page 169.

  Important: The input file will be deleted after the parameter is merged with the default xml file.

-output
: The output file contains further information. It will be written into the "/tmp" directory. In the description of each system action there is an example output file.

Action “createsystem”

This action causes a new system to be created. This new system is a copy of the active system.

Syntax

```
►►──rc = CB390CFG──(″── -action──’createsystem’ ──────────────────────────►
►─ -xmlinput─'defaultxmlfilename' ──┬──────────────────────────┬──►
                                     └─ -input─'inputfilename' ─┘

►─ -output─'outputfilename' ─")─────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for createsystem "inputcreatesystem.xml" is listed in section "inputcreatesystem.xml" on page 189. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `'inputcreatesystem.xml`. Otherwise specify the complete location to the default xml file by setting this parameter to `'/usr/lpp/WebSphere/samples/smapi/inputcreatesystem.xml'`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this system action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputcreatesystem.xml" on page 189.

Parameter name	Values	Required
conversationname	Name of the conversation	x

Parameter name	Values	Required
systemname	Name of the system	x
systemdescription	Description of the system	

Example

Here is an example script:

```
/* REXX function */
/* Functiontest Test  createsystem*/
/* Dependencies: */
/* The conversation "API Functiontest" must be added*/
/* The system "SY2" must not be added in the conversation "API Functiontest"*/
call syscalls 'ON'
signal on error

say "FCT Test #createsystem"

name. = 0
name.1  = "conversationname"
name.2  = "systemname"

val. = 0
val.1 = "API Functiontest"
val.2 = "SY2"
rc = 4
i = 1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #createsystem failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: createsystem */
rc = CB390CFG("-action 'createsystem' -xmlinput 'inputcreatesystem.xml'
-input 'tempin' -output 'createsystem'")
if (rc == 4) then do
  say "FCT Test #createsystem failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("createsystem"  "ALL")
if (rc == 4) then do
  say "FCT Test #createsystem failed while XMLPARSE"
  exit
end
say "FCT Test #createsystem completed"
return 0
```

```
exit
error:
say "Error in FCT Test #createsystem" rc "at line" sigl
say sourceline(sigl)
exit
```

Action “deletesystem”

This action causes the named system to be deleted.

Syntax

```
►►──rc = CB390CFG──("── -action──'deletesystem' ──────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬──────────────────────────┬────────►
                                     └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")──────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for deletesystem "inputdeletesystem.xml" is listed in section “inputdeletesystem.xml” on page 191. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputdeletesystem.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputdeletesystem.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these new specified values. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file
The table below includes all of the attributes that are known for this conversation action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputdeletesystem.xml" on page 191.

Parameter name	Values	Required
conversationname	Name of the conversation	x
systemname	Name of the system	x

Example
Here is an example script:

```
/* REXX function */
/* Functiontest Test 09: deletesystem*/

/* Dependencies: */
/* The conversation "API Functiontest" must be added*/
/* The system "SY2" must be added in the conversation "API Functiontest"*/
call syscalls 'ON'
signal on error

say "FCT Test #deletesystem"

name. = 0
name.1 = "conversationname"
name.2 = "systemname"

val. = 0
val.1 = "API Functiontest"
val.2 = "SY2"

rc = 4
i = 1

/*Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #deletesystem failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: deletesystem */
rc = CB390CFG("-action 'deletesystem' -xmlinput 'inputdeletesystem.xml'
-input 'tempin' -output 'deletesystem'")
if (rc == 4) then do
  say "FCT Test #deletesystem failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("deletesystem"  "ALL")
if (rc == 4) then do
  say "FCT Test #deletesystem failed while XMLPARSE"
  exit
end
say "FCT Test #deletesystem completed"
exit

error:
say "Error in FCT Test #deletesystem" rc "at line" sigl
say sourceline(sigl)
exit
```

Action “changesystem”

This causes attributes of the named system to be changed.

Syntax

```
►►──rc = CB390CFG──("── -action──'changesystem' ──────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬─────────────────────────┬──────────►
                                     └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")──────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for changesystem "inputchangesystem.xml" is listed in section “inputchangesystem.xml” on page 190. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename "`inputchangesystem.xml`". Otherwise specify the complete location to the default xml file by setting this parameter to "`/usr/lpp/WebSphere/samples/smapi/inputchangesystem.xml`". If

| you want to use your own default xml file you **must** specify the
| complete directory of the file or you **must** set the
| `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this system action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputchangesystem.xml" on page 190.

Parameter name	Values	Required
conversationname	Name of the conversation	x
systemname	Name of the system	x
systemdecription	Description of the system	

Example

Here is an example script:

```
/* REXX function */
/* Functiontest Test : changesystem*/
/* Dependencies: */
/* The conversation "API Functiontest" must be added*/
/* The system "SY2" must be added in the conversation "API Functiontest"*/
call syscalls 'ON'
signal on error

say "FCT Test #changesystem"

name. = 0
name.1  = "conversationname"
name.2  = "systemname"
name.3  = "systemdescription"

val. = 0
val.1  = "API Functiontest"
val.2  = "SY2"
val.3  = "New description"
```

```
rc = 4
i = 1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #changesystem failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: changeesystem */
rc = CB390CFG("-action 'changesystem' -xmlinput 'inputchangesystem.xml'
-input 'tempin' -output 'changesystem'")
if (rc == 4) then do
  say "FCT Test #changesystem failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("changesystem"  "ALL")
if (rc == 4) then do
  say "FCT Test #changesystem failed while XMLPARSE"
  exit
end
say "FCT Test #changesystem completed"
return 0
exit

error:
say "Error in FCT Test #changesystem" rc "at line" sigl
say sourceline(sigl)
exit
```

Action “listsystem”

This causes the named systems to be listed. If the system name equals "*" then all systems will be listed.

Syntax

```
►►──rc = CB390CFG──("── -action──'listsystem' ─────────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬─────────────────────────────┬─────────►
                                    └─ -input──'inputfilename' ───┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for listsystem "inputlistsystem.xml" is listed in section "inputlistsystem.xml" on page 191. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

  If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputlistsystem.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputlistsystem.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
: This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
: This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this system action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputlistsystem.xml" on page 191.

Parameter name	Values	Required
conversationname	Name of the conversation "*" to list all conversations pertaining to the Administrator	x
systemname	Name of the system	x

Example

Here is an example script:

```
/* REXX function */
/* Functiontest Test listsystem: listsystem*/

/* Dependencies: */
/* For Part 1: "List special system" the system "SY2" must be created */
/*             conversation "API Functiontest" must be added */
/* For Part 2: "List all systems" none */
call syscalls 'ON'
signal on error

say "FCT Test #listsystem"

name. = 0
name.1 = "conversationname"
name.2 = "systemname"

/* Part 1: */
/* List special system */

val. = 0
val.1 = "API Functiontest"
val.2 = "SY1"

rc = 4
i = 1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #listsystem failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: listsystem */
rc = CB390CFG("-action 'listsystem' -xmlinput 'inputlistsystem.xml'
-input 'tempin' -output 'listsystem'")
if (rc == 4) then do
  say "FCT Test #listsystem failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("listsystem" "ALL")
if (rc == 4) then do
  say "FCT Test #listsystem failed while XMLPARSE"
  exit
end

/* Part 2: */
/* List all systems */

val.2 = "*"
```

```
i=1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if rc == 4 then do
    say "FCT Test #listsystem failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: listsystem */
rc = CB390CFG("-action 'listsystem' -xmlinput 'inputlistsystem.xml'
-input 'tempin' -output 'listsystemB'")
if rc == 4 then do
  say "FCT Test #listsystem failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("listsystemB" "ALL")
if rc == 4 then do
  say "FCT Test #listsystem failed while XMLPARSE"
  exit
end
say "FCT Test #listsystem completed"
exit

error:
say "Error in FCT Test #listsystem" rc "at line" sigl
say sourceline(sigl)
exit
```

Server

These functions are for the modifications of a server.

Syntax

```
►►──rc = CB390CFG──("── -action──'──┬─create-  ─┬─server──'──────────────►
                                   ├─delete-  ─┤
                                   ├─change-  ─┤
                                   └─list-  ───┘

►─ -xmlinput──'defaultxmlfilename' ─┬────────────────────────────┬───────►
                                    └─ -input──'inputfilename' ──┘

►─ -output──'outputfilename' ──")─────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "4" an error has occurred while processing the action.

action

createserver	Causes a new server to be created.
deleteserver	Causes the server to be deleted.
changeserver	Causes the server to be changed.
listserver	Causes the server to be listed.

-xmlinput
This is the default xml file. In this file all required parameters for the action which should be performed **must** be specified. This file is a xml file with a document type definition (DTD). The DTD only specifies the structure of the document. The user can specify default values for each parameter. These parameters can be overriden by the REXX script via the input parameter. All default xml files are listed in "Chapter 9. Default XML files" on page 185. The parameters in these files are set to the default values of the SM Administration EUI.

The default xml file **must** be present in the path that is specified by the environment variable `DEFAULT_CLIENT_XML_PATH` or the user has to specify its path.

Example: `-xmlinput 'inputcreateserver.xml'` specifies the default input xml file in the `DEFAULT_CLIENT_XML_PATH`. But `-xmlinput './inputcreateserver.xml'` specifies the file in the current directory.

The user can modify the default path for the default xml files by setting the environment variable `DEFAULT_CLIENT_XML_PATH` to another existing path. Make sure that the path exists and that the default xml file that will be used is in this directory.

-input
This is an optional parameter for the CB390CFG API. It specifies the input file which contains the name value pairs that should override the parameters of the default xml file. To generate a xml file by using REXX variables there is a tool called XMLGEN. This tool is described in "Chapter 5. XMLGEN" on page 169.

Important: The input file will be deleted after the parameters are merged with the default xml file.

-output
The output file contains further information. In the description of

each server action there is an example output file. The general output format for a server action looks like this:

```
administratorname.1 AdministratorName
allownonauthenticatedclients.1 Y|N
allowserverregiongarbagecollection.1 Y|N
allowssl.1 Y|N
allowuseridpasswd.1 Y|N
conversationname.1 ConversationName
dcekeytabfile.1 DCEKeyTabFile
dcequalityofprotection.1 DCEQualityOfProtectionState
debuggerallowed.1 Y|N
garbagecollectioninterval.1 Number(0-2G)
identityofthecontrolregion.1 IdentityOfTheControlRegion
identityoftheserverregion.1 IdentityOfTheServerRegion
isolationpolicy.1 IsolationPolicyState
localidentity.1 LocalIdentity
logstreamname.1 LogStreamName
procname.1 ProcName
productionserver.1 Y|N
remoteidentity.1 RemoteIdentity
replicationpolicy.1 ReplicationPolicyStata
serverdescription.1 ServerDescription
servername.1 ServerName
serverregionjvmname.1 ServerRegionJVMName
serverregionrequiresjvm.1 Y|N
serverregionstacksize.1 Number(0-100000)
sslracfkeyring.1 SSL_RACF_Keyring
sslv2timeout.1 SSL_V2Timeout
sslv3timeout.1 SSL_V3Timeout
sysplexname.1 SysplexName
transactionfactory.1 Y|N
usedce.1 Y|N
useridpassticket.1 Y|N
security.1 Security
status 0|4
message.1 OK|ErrorMessage
count NumberOfListedServers
```

Action “createserver”

This action causes a new server to be created.

Syntax

```
►►──rc = CB390CFG──("── -action──'createserver' ──────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬──────────►
                                    └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")─────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for createserver "inputcreateserver.xml" is listed in section "inputcreateserver.xml" on page 192. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputcreateserver.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputcreateserver.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this server action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputcreateserver.xml" on page 192.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	x
serverdescription	Description of the server	
identityofthecontrolregion		x
identityoftheserverregion		x
serverregionstacksize	Numerical	x

Parameter name	Values	Required
productionserver	Allowed values are "Y" or "N"	x
debuggerallowed	Allowed values are "Y" or "N"	x
isolationpolicy	Allowed values are `One_Transaction_Per_Server_Region` `Multiple_Transactions_Per_Server_Region`	x
replicationpolicy	Allowed values are `One_Per_Server` `Replicate_As_Needed`	x
serverregionrequiresjvm	Allowed values are "Y" or "N"	x
serverregionjvmname	Name of the JVM	x
localidentity	Name of the local identity	x
remoteidentity	Name of the remote identity	x
transactionfactory	Allowed values are "Y" or "N"	x
allowserverregiongarbagecollection	Allowed values are "Y" or "N"	x
garbagecollectioninterval	Numerical value between 0 and 2G	x
logstreamname	Name of the logstream	x
procname		x
allownonathenticatedclients	Allowed values are "Y" or "N"	x
allowuseridpasswd	Allowed values are "Y" or "N"	x
useridpassticket	Allowed values are "Y" or "N"	x
usedce	Allowed values are "Y" or "N"	x
dcequalityofprotection	Allowed Values are `No_Protection` `Integrity` `Confidentiality`	x
dcekeytabfile	Name of the DCE Keytab File	x
allowssl	Allowed values are "Y" or "N"	x
sslracfkeyring		x
sslv2timeout	Numeric value between 1 and 100	x
sslv3timeout	Numeric value between 1 and 86400	x

Parameter name	Values	Required

security	Preference Of Security Type
	This specifies the sequence of the security. To specify security in the default xml file there must be set an element for each security type. If a security type is specified the attribute must be set.
	To specify in REXX only one element must be specified. To set the security type `security value [value value ...]` where `value` specifies the value of the Security type. If security is specified at least one value **must** be set. Allowed values are
	`ISM_DCE`
	`ISM_UserID_Password`
	`ISM_Pass_Ticket`
	`ISM_SSL`	

smfwrserveractivity	Allowed values are "Y" or "N"	x
smfwrcontaineractivity	Allowed values are "Y" or "N"	x
smfwrserverinterval	Allowed values are "Y" or "N"	x
smfwrcontainerinterval	Allowed values are "Y" or "N"	x
smfintervallength	Numeric value between 15 and 86400 or 0	x
environment	To specify the environment in the default xml there must be set an element for each environment type. If an environment type is specified the attributes **must** be set. To specify the environment in REXX only one element must be specified. To set the environment type `environment name ='value' [name ='value' ...]`, where `name` specifies the environment name and value the value of the environment. If the environment is specified atleast one `name ='value'` pair **must** be specified.	
allowsslclientcerts	Allowed values are "Y" or "N"	x
olthostname	Character (256)	x
oltport	Character value between 1 and 65535	x
acceptassertedid	Allowed values are "Y" or "N"	x
allowkerberos	Allowed values are "Y" or "N"	x
sendassertedid	Allowed values are "Y" or "N"	x

For the server properties some changes have occured between the SM-EUI and the Scripting API. Below there is a table of the different values.

Parameter for "DCE Quality Of Protection"

Script value	GUI value
Integrity	Message Integrity
Confidentiality	Message Confidentiality

Example

Here is an example script:

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1  = "conversationname"
name.2  = "servername"
name.3  = "identityofthecontrolregion"
name.4  = "identityoftheserverregion"
name.5  = "serverregionstacksize"
name.6  = "productionserver"
name.7  = "debuggerallowed"
name.8  = "isolationpolicy"
name.9  = "replicationpolicy"
name.10 = "serverregionrequiresjvm"
name.11 = "serverregionjvmname"
name.12 = "localidentity"
name.13 = "remoteidentity"
name.14 = "transactionfactory"
name.15 = "allowserverregiongarbagecollection"
name.16 = "garbagecollectioninterval"
name.17 = "logstreamname"
name.18 = "procname"
name.19 = "allownonauthenticatedclients"
name.20 = "allowuseridpasswd"
name.21 = "useridpassticket"
name.22 = "usedce"
name.23 = "dcequalityofprotection"
name.24 = "dcekeytabfile"
name.25 = "security"
name.26 = "allowssl"
name.27 = "serverdescription"
name.28 = "sslracfkeyring"
name.29 = "sslv2timeout"
name.30 = "sslv3timeout"

val. = 0
val.1  = "API Functiontest"
val.2  = "APIFCT"
val.3  = "IBMUSER"
val.4  = "IBMUSER"
val.5  = "0"
val.6  = "Y"
val.7  = "N"
val.8  = "Multiple_Transactions_Per_Server_Region"
val.9  = "One_Per_Server"
val.10 = "N"
val.11 = ""
val.12 = "CBGUEST"
val.13 = "CBGUEST"
val.14 = "N"
val.15 = "Y"
val.16 = "50000"
val.17 = ""
val.18 = "BBOASR1"
val.19 = "Y"
val.20 = "Y"
val.21 = "N"
val.22 = "N"
val.23 = "No_Protection"
val.24 = ""
val.25 = "ISM_UserID_Password"
val.26 = "N"
val.27 = "APIFCT Description"
val.28 = ""
val.29 = "100"
val.30 = "600"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #07 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'createserver' -xmlinput 'inputcreateserver.xml'
               -input 'tempin' -output 'FCT07'")
if (rc == 4) then do
  say "FCT Test #07 failed"
  exit
end
exit

error:
say "Error in FCT Test #07" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
allownonauthenticatedclients.1 Y
allowserverregiongarbagecollection.1 Y
allowssl.1 N
allowuseridpasswd.1 Y
conversationname.1 API Functiontest
dcekeytabfile.1
dcequalityofprotection.1 No_Protection
debuggerallowed.1 N
garbagecollectioninterval.1 50000
identityofthecontrolregion.1 IBMUSER
identityoftheserverregion.1 IBMUSER
isolationpolicy.1 Multiple_Transactions_Per_Server_Region
localidentity.1 CBGUEST
logstreamname.1
procname.1 BBOASR1
productionserver.1 Y
remoteidentity.1 CBGUEST
replicationpolicy.1 One_Per_Server
serverdescription.1 APIFCT Description
servername.1 APIFCT
serverregionjvmname.1
serverregionrequiresjvm.1 N
serverregionstacksize.1 0
sslracfkeyring.1
sslv2timeout.1
sslv3timeout.1
sysplexname.1 PLEX1
transactionfactory.1 N
usedce.1 N
useridpassticket.1 N
security.1 ISM_UserID_Password
environment.1 CLASSPATH = 'Demo:test1' PATH = 'test2'
status 0
message.1 OK
count 1
```

Action “deleteserver”

This action causes a server to be deleted. This is a logical deletion. The deletion does not actually occur until the conversation this change is associated with is commited.

Syntax

```
►►──rc = CB390CFG──("── -action──'deleteserver' ──────────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬─────────────────────────────┬──────────►
                                     └─ -input──'inputfilename' ───┘
►─ -output─'outputfilename' ─")──────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for deleteserver "inputdeleteserver.xml" is listed in section "inputdeleteserver.xml" on page 194. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputdeleteserver.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputdeleteserver.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this server action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputdeleteserver.xml" on page 194.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	x

Example

Here is an example script:

```
/* REXX function */
call syscalls 'ON'
signal on error

say "FCT Test #09"

name. = 0
name.1 = "conversationname"
name.2 = "servername"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #09 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'deleteserver' -xmlinput 'inputdeleteserver.xml'
              -input 'tempin' -output 'FCT09'")
if (rc == 4) then do
  say "FCT Test #09 failed"
  exit
end
exit

error:
say "Error in FCT Test #09" rc "at line" sigl
say sourceline(sigl)
exit
```

```
administratorname.1 CBADMIN
allownonauthenticatedclients.1 Y
allowserverregiongarbagecollection.1 Y
allowssl.1 N
allowuseridpasswd.1 Y
conversationname.1 API Functiontest
dcekeytabfile.1
dcequalityofprotection.1 No_Protection
debuggerallowed.1 N
garbagecollectioninterval.1 50000
identityofthecontrolregion.1 IBMUSER
identityoftheserverregion.1 IBMUSER
isolationpolicy.1 Multiple_Transactions_Per_Server_Region
localidentity.1 CBGUEST
```

```
logstreamname.1
procname.1 BBOASR1
productionserver.1 Y
remoteidentity.1 CBGUEST
replicationpolicy.1 One_Per_Server
serverdescription.1 APIFCT Description
servername.1 APIFCT
serverregionjvmname.1
serverregionrequiresjvm.1 N
serverregionstacksize.1 0
sslracfkeyring.1
sslv2timeout.1 0
sslv3timeout.1 0
sysplexname.1 PLEX1
transactionfactory.1 N
usedce.1 N
useridpassticket.1 N
security.1 ISM_UserID_Password ISM_DCE
environment.1 CLASSPATH = 'testchange1' PATH = 'testchange2'
status 0
message.1 OK
count 1
```

Action “changeserver”

This action causes attributes of the named server to be changed.

Syntax

```
►►──rc = CB390CFG──("── -action──'changeserver' ──────────────────────────►

►── -xmlinput──'defaultxmlfilename' ──┬──────────────────────────┬────────►
                                      └── -input──'inputfilename' ──┘

►── -output──'outputfilename' ──")──────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for changeserver "inputchangeserver.xml" is listed in section “inputchangeserver.xml” on page 195. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename

| "`inputchangeserver.xml`". Otherwise specify the complete location
| to the default xml file by setting this parameter to
| "`/usr/lpp/WebSphere/samples/smapi/inputchangeserver.xml`". If
| you want to use your own default xml file you **must** specify the
| complete directory of the file or you **must** set the
| `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file
The table below includes all of the attributes that are known for this server action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputchangeserver.xml" on page 195.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	x
serverdescription	Description of the server	
identityofthecontrolregion		x
identityoftheserverregion		x
serverregionstacksize	Numerical	x
productionserver	Allowed values are "Y" or "N"	x
debuggerallowed	Allowed values are "Y" or "N"	x
isolationpolicy	Allowed values are `One_Transaction_Per_server_Region` `Multiple_Transactions_Per_Server_Region`	x
replicationpolicy	Allowed values are `One_Per_Server` `Replicate_As_Needed`	x
serverregionrequiresjvm	Allowed values are "Y" or "N"	x
serverregionjvmname	Name of the JVM	x

Parameter name	Values	Required
localidentity	Name of the local identity	x
remoteidentity	Name of the remote identity	x
transactionfactory	Allowed values are "Y" or "N"	x
allowserverregiongarbagecollection	Allowed values are "Y" or "N"	x
garbagecollectioninterval	Numerical value between 0 and 2G	x
logstreamname	Name of the logstream	x
procname		x
allownonathenticatedclients	Allowed values are "Y" or "N"	x
allowuseridpasswd	Allowed values are "Y" or "N"	x
useridpassticket	Allowed values are "Y" or "N"	x
usedce	Allowed values are "Y" or "N"	x

dcequalityofprotection	Allowed values are
	`No_Protection`
	`Integrity`
	`Confidentiality`
		x

dcekeytabfile	Name of the DCE Keytab File	x
allowssl	Allowed values are "Y" or "N"	x
sslracfkeyring		x
sslv2timeout	Numeric value between 1 and 100	x
sslv3timeout	Numeric value between 1 86400	x

security	Preference Of Security Type
	This specifies the sequence of the security. To specify security in the default xml file there must be set an element for each security type. If a security type is specified the attribute **must** be set.
	To specify in REXX only one element must be specified. To set the security type `security value [value value ...]` where `value` specifies the value of the Security type. If security is specified at least one value **must** be set. Allowed values are
	`ISM_DCE`
	`ISM_UserID_Password`
	`ISM_Pass_Ticket`
	`ISM_SSL`	

smfwrserveractivity	Allowed values are "Y" or "N"	x
smfwrcontaineractivity	Allowed values are "Y" or "N"	x
smfwrserverinterval	Allowed values are "Y" or "N"	x

Parameter name	Values	Required
smfwrcontainerinterval	Allowed values are "Y" or "N"	x
smfintervallength	Numeric value between 15 and 86400 or 0	x
environment	To specify the environment in the default xml there must be set an element for each environment type. If an environment type is specified the attributes **must** be set. To specify the environment in REXX only one element must be specified. To set the environment type `environment name ='value' [name ='value' ...]`, where `name` specifies the environment name and value the value of the environment. If the environment is specified at least one `name ='value'` pair **must** be specified.	
allowsslclientcerts	Allowed values are "Y" or "N"	x
olthostname	Character (256)	x
oltport	Character value between 1 and 65535	x
acceptassertedid	Allowed values are "Y" or "N"	x
allowkerberos	Allowed values are "Y" or "N"	x
sendassertedid	Allowed values are "Y" or "N"	x

For the server properties some changes have occured between the SM-EUI and the Scripting API. Below there is a table of the different values.

Parameter for "DCE Quality Of Protection"

Script value	GUI value
Integrity	Message Integrity
Confidentiality	Message Confidentiality

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1  = "conversationname"
name.2  = "servername"
name.3  = "identityofthecontrolregion"
name.4  = "identityoftheserverregion"
name.5  = "serverregionstacksize"
name.6  = "productionserver"
name.7  = "debuggerallowed"
name.8  = "isolationpolicy"
name.9  = "replicationpolicy"
name.10 = "serverregionrequiresjvm"
name.11 = "serverregionjvmname"
name.12 = "localidentity"
name.13 = "remoteidentity"
name.14 = "transactionfactory"
name.15 = "allowserverregiongarbagecollection"
name.16 = "garbagecollectioninterval"
name.17 = "logstreamname"
name.18 = "procname"
name.19 = "allownonauthenticatedclients"
name.20 = "allowuseridpasswd"
name.21 = "useridpassticket"
name.22 = "usedce"
name.23 = "dcequalityofprotection"
name.24 = "dcekeytabfile"
name.25 = "security"
name.26 = "allowssl"
name.27 = "serverdescription"
name.28 = "sslracfkeyring"
name.29 = "sslv2timeout"
name.30 = "sslv3timeout"

val. = 0
val.1  = "API Functiontest"
val.2  = "APIFCT"
val.3  = "IBMUSER"
val.4  = "IBMUSER"
val.5  = "0"
val.6  = "Y"
val.7  = "N"
val.8  = "Multiple_Transactions_Per_Server_Region"
val.9  = "One_Per_Server"
val.10 = "N"
val.11 = ""
val.12 = "CBGUEST"
val.13 = "CBGUEST"
val.14 = "N"
val.15 = "Y"
val.16 = "50000"
val.17 = ""
val.18 = "BBOASR1"
val.19 = "Y"
val.20 = "Y"
val.21 = "N"
val.22 = "N"
val.23 = "No_Protection"
val.24 = ""
val.25 = "ISM_UserID_Password"
val.26 = "N"
val.27 = "APIFCT Description"
val.28 = ""
val.29 = "200"
val.30 = "500"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #08 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'changeserver' -xmlinput 'inputchangeserver.xml'
               -input 'tempin' -output 'FCT08'")
if (rc == 4) then do
  say "FCT Test #08 failed"
  exit
end
exit

error:
say "Error in FCT Test #08" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
allownonauthenticatedclients.1 Y
allowserverregiongarbagecollection.1 Y
allowssl.1 N
allowuseridpasswd.1 Y
conversationname.1 API Functiontest
dcekeytabfile.1
dcequalityofprotection.1 No_Protection
debuggerallowed.1 N
garbagecollectioninterval.1 50000
identityofthecontrolregion.1 IBMUSER
identityoftheserverregion.1 IBMUSER
isolationpolicy.1 Multiple_Transactions_Per_Server_Region
localidentity.1 CBGUEST
logstreamname.1
procname.1 BBOASR1
productionserver.1 Y
remoteidentity.1 CBGUEST
replicationpolicy.1 One_Per_Server
serverdescription.1 APIFCT Description
servername.1 APIFCT
serverregionjvmname.1
serverregionrequiresjvm.1 N
serverregionstacksize.1 0
```

```
sslracfkeyring.1
sslv2timeout.1
sslv3timeout.1
sysplexname.1 PLEX1
transactionfactory.1 N
usedce.1 N
useridpassticket.1 N
security.1 ISM_UserID_Password ISM_DCE
environment.1 CLASSPATH = 'testchange1' PATH = 'testchange2'
status 0
message.1 OK
count 1
```

Action “listserver”

This action causes the named server to be listed. If the server name equals "*" then all servers will be listed.

Syntax

```
►►──rc = CB390CFG──("── -action──'listserver' ──────────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬──────────────────────────┬──────────►
                                     └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for listserver "inputlistserver.xml" is listed in section “inputlistserver.xml” on page 197. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory. If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputlistserver.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputlistserver.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
: This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
: This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this server action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputlistserver.xml" on page 197.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	"*" to list all server in the conversation	x

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "servername"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #10 failed while XMLGEN"
    exit
  end
  i = i+1
end;
```

```
rc = CB390CFG("-action 'listserver' -xmlinput 'inputlistserver.xml'
               -input 'tempin' -output 'FCT10'")
if (rc == 4) then do
  say "FCT Test #10 failed"
  exit
end
exit

error:
say "Error in FCT Test #10" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
allownonauthenticatedclients.1 Y
allowserverregiongarbagecollection.1 Y
allowssl.1 N
allowuseridpasswd.1 Y
conversationname.1 API Functiontest
dcekeytabfile.1
dcequalityofprotection.1 No_Protection
debuggerallowed.1 N
garbagecollectioninterval.1 50000
identityofthecontrolregion.1 IBMUSER
identityoftheserverregion.1 IBMUSER
isolationpolicy.1 Multiple_Transactions_Per_Server_Region
localidentity.1 CBGUEST
logstreamname.1
procname.1 BBOASR1
productionserver.1 Y
remoteidentity.1 CBGUEST
replicationpolicy.1 One_Per_Server
serverdescription.1 APIFCT Description
servername.1 APIFCT
serverregionjvmname.1
serverregionrequiresjvm.1 N
serverregionstacksize.1 0
sslracfkeyring.1
sslv2timeout.1 0
sslv3timeout.1 0
sysplexname.1 PLEX1
transactionfactory.1 N
usedce.1 N
useridpassticket.1 N
security.1 ISM_UserID_Password ISM_DCE
environment.1 CLASSPATH = 'testchange1' PATH = 'testchange2'
status 0
message.1 OK
count 1
```

J2EE Server

These functions are for the modifications of a J2EE server.

Syntax

```
►►──rc = CB390CFG──("── -action──'──┬─create- ─┬──server──'──────────────────►
                                    ├─delete- ─┤
                                    ├─change- ─┤
                                    └─list-  ──┘

►── -xmlinput──'defaultxmlfilename' ──┬─────────────────────────────┬─────────►
                                      └─ -input──'inputfilename' ──┘

►── -output──'outputfilename' ──")─────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "4" an error has occurred while processing the action.

action

createj2eeserver	Causes a new J2EE server to be created.
deletej2eeserver	Causes the J2EE server to be deleted.
changej2eeserver	Causes the J2EE server to be changed.
listj2eeserver	Causes the J2EE server to be listed.

-xmlinput
This is the default xml file. In this file all required parameters for the action which should be performed **must** be specified. This file is a xml file with a document type definition (DTD). The DTD only specifies the structure of the document. The user can specify default values for each parameter. These parameters can be overriden by the REXX script via the input parameter. All default xml files are listed in "Chapter 9. Default XML files" on page 185. The parameters in these files are set to the default values of the SM Administration EUI.

The default xml file **must** be present in the path that is specified by the environment variable `DEFAULT_CLIENT_XML_PATH` or the user has to specify its path.

Example: `-xmlinput 'inputcreatej2eeserver.xml'` specifies the default input xml file in the `DEFAULT_CLIENT_XML_PATH`. But `-xmlinput './inputcreatej2eeserver.xml'` specifies the file in the current directory.

The user can modify the default path for the default xml files by setting the environment variable `DEFAULT_CLIENT_XML_PATH` to another existing path. Make sure that the path exists and that the default xml file that will be used is in this directory.

-input
This is an optional parameter for the CB390CFG API. It specifies the input file which contains the name value pairs that should override the parameters of the default xml file. To generate a xml file by using REXX variables there is a tool called XMLGEN. This tool is described in "Chapter 5. XMLGEN" on page 169.

Important: The input file will be deleted after the parameters are merged with the default xml file.

-output
The output file contains further information. In the description of each J2EE server action there is an example output file. The general output format for a J2EE server action looks like this:

```
administratorname.1 AdministratorName
allownonauthenticatedclients.1 Y|N
allowserverregiongarbagecollection.1 Y|N
allowssl.1 Y|N
allowuseridpasswd.1 Y|N
conversationname.1 ConversationName
dcekeytabfile.1 DCEKeyTabFile
dcequalityofprotection.1 DCEQualityOfProtectionState
debuggerallowed.1 Y|N
garbagecollectioninterval.1 Number(0-2G)
identityofthecontrolregion.1 IdentityOfTheControlRegion
identityoftheserverregion.1 IdentityOfTheServerRegion
isolationpolicy.1 IsolationPolicyState
localidentity.1 LocalIdentity
logstreamname.1 LogStreamName
procname.1 ProcName
productionserver.1 Y|N
remoteidentity.1 RemoteIdentity
replicationpolicy.1 ReplicationPolicyStata
serverdescription.1 ServerDescription
servername.1 ServerName
serverregionjvmname.1 ServerRegionJVMName
serverregionrequiresjvm.1 Y|N
serverregionstacksize.1 Number(0-100000)
sslracfkeyring.1 SSL_RACF_Keyring
sslv2timeout.1 SSL_V2Timeout
sslv3timeout.1 SSL_V3Timeout
sysplexname.1 SysplexName
```

```
transactionfactory.1 Y|N
usedce.1 Y|N
useridpassticket.1 Y|N
security.1 Security
status 0|4
message.1 OK|ErrorMessage
count NumberOfListedServers
```

Action “createj2eeserver”

This action causes a new J2EE server to be created.

Syntax

```
►►──rc = CB390CFG──("── -action──'createj2eeserver' ──────────────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬──────────►
                                    └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")──────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for createj2eeserver "inputcreatej2eeserver.xml" is listed in section “inputcreatej2eeserver.xml” on page 201. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

  If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputcreatej2eeserver.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputcreatej2eeserver.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
: This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these

new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file
The table below includes all of the attributes that are known for this server action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputcreatej2eeserver.xml" on page 201.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the J2EE server	x
serverdescription	Description of the J2EE server	
identityofthecontrolregion		x
identityoftheserverregion		x
serverregionstacksize	Numerical	x
productionserver	Allowed values are "Y" or "N"	x
debuggerallowed	Allowed values are "Y" or "N"	x
isolationpolicy	Allowed values are `One_Transaction_Per_Server_Region` `Multiple_Transactions_Per_Server_Region`	x
replicationpolicy	Allowed values are `One_Per_Server Replicate_As_Needed`	x
serverregionrequiresjvm	Allowed values are "Y" or "N"	x
serverregionjvmname	Name of the JVM	x
localidentity	Name of the local identity	x
remoteidentity	Name of the remote identity	x
transactionfactory	Allowed values are "Y" or "N"	x
allowserverregiongarbagecollection	Allowed values are "Y" or "N"	x
garbagecollectioninterval	Numerical value between 0 and 2G	x
logstreamname	Name of the logstream	x
procname		x
allownonathenticatedclients	Allowed values are "Y" or "N"	x

Parameter name	Values	Required
allowuseridpasswd	Allowed values are "Y" or "N"	x
useridpassticket	Allowed values are "Y" or "N"	x
usedce	Allowed values are "Y" or "N"	x

dcequalityofprotection	Allowed Values are
	`No_Protection`
	`Integrity`
	`Confidentiality`
		x

dcekeytabfile	Name of the DCE Keytab File	x
allowssl	Allowed values are "Y" or "N"	x
sslracfkeyring		x
sslv2timeout	Numeric value between 1 and 100	x
sslv3timeout	Numeric value between 1 and 86400	x

security	Preference Of Security Type
	This specifies the sequence of the security. To specify security in the default xml file there must be set an element for each security type. If a security type is specified the attribute must be set.
	To specify in REXX only one element must be specified. To set the security type `security value [value value ...]` where `value` specifies the value of the Security type. If security is specified at least one value **must** be set. Allowed values are
	`ISM_DCE`
	`ISM_UserID_Password`
	`ISM_Pass_Ticket`
	`ISM_SSL`	

smfwrserveractivity	Allowed values are "Y" or "N"	x
smfwrcontaineractivity	Allowed values are "Y" or "N"	x
smfwrserverinterval	Allowed values are "Y" or "N"	x
smfwrcontainerinterval	Allowed values are "Y" or "N"	x
smfintervallength	Numeric value between 15 and 86400 or 0	x

Parameter name	Values	Required
environment	To specify the environment in the default xml there must be set an element for each environment type. If an environment type is specified the attributes **must** be set. To specify the environment in REXX only one element must be specified. To set the environment type `environment name ='value' [name ='value' ...]`, where `name` specifies the environment name and value the value of the environment. If the environment is specified atleast one `name ='value'` pair **must** be specified.	
allowsslclientcerts	Allowed values are "Y" or "N"	x
olthostname	Character (256)	x
oltport	Character value between 1 and 65535	x
acceptassertedid	Allowed values are "Y" or "N"	x
allowkerberos	Allowed values are "Y" or "N"	x
sendassertedid	Allowed values are "Y" or "N"	x

For the J2EE server properties some changes have occured between the SM-EUI and the Scripting API. Below there is a table of the different values.

Parameter for "DCE Quality Of Protection"

Script value	GUI value
Integrity	Message Integrity
Confidentiality	Message Confidentiality

Example

Here is an example script:

```
/* REXX function */
/* Functiontest Test : createj2eeserver*/
/* Dependencies: */
/* The conversation "API Functiontest" must be added*/
/* The J2EEserver "J2EESRV" must not be added in the conversation
"API Functiontest"*/
call syscalls 'ON'
signal on error

say "FCT Test #createj2eeserver"

name. = 0
name.1  = "conversationname"
name.2  = "j2eeservername"
name.3  = "identityofthecontrolregion"
name.4  = "identityoftheserverregion"
name.5  = "serverregionstacksize"
name.6  = "productionserver"
name.7  = "debuggerallowed"
name.8  = "isolationpolicy"
name.9  = "replicationpolicy"
name.10 = "serverregionrequiresjvm"
name.11 = "serverregionjvmname"
name.12 = "localidentity"
name.13 = "remoteidentity"
name.14 = "transactionfactory"
name.15 = "allowserverregiongarbagecollection"
name.16 = "garbagecollectioninterval"
name.17 = "logstreamname"
name.18 = "procname"
name.19 = "allownonauthenticatedclients"
name.20 = "allowuseridpasswd"
name.21 = "useridpassticket"
name.22 = "usedce"
name.23 = "dcequalityofprotection"
name.24 = "dcekeytabfile"
name.25 = "security"
name.26 = "allowssl"
name.27 = "j2eeserverdescription"
name.28 = "sslracfkeyring"
name.29 = "sslv2timeout"
name.30 = "sslv3timeout"
name.31 = "environment"
name.32 = "allowsslclientcerts"
name.33 = "olthostname"
name.34 = "oltport"
name.35 = "allowkerberos"
name.36 = "acceptassertedid"
name.37 = "sendassertedid"

val. = 0
val.1  = "API Functiontest"
val.2  = "J2EESRV"
val.3  = "IBMUSER"
val.4  = "IBMUSER"
val.5  = "0"
val.6  = "Y"
val.7  = "N"
val.8  = "Multiple_Transactions_Per_Server_Region"
val.9  = "One_Per_Server"
val.10 = "N"
val.11 = ""
val.12 = "CBGUEST"
val.13 = "CBGUEST"
val.14 = "N"
val.15 = "Y"
val.16 = "50000"
val.17 = ""
val.18 = "BBOASR1"
val.19 = "Y"
val.20 = "Y"
val.21 = "N"
val.22 = "N"
val.23 = "No_Protection"
val.24 = ""
val.25 = "ISM_UserID_Password"
val.26 = "N"
val.27 = "APIFCT Description"
val.28 = ""
val.29 = "100"
val.30 = "600"
val.31 = "CLASSPATH='Demo:test1' PATH='test2'
DEFAULT_CLIENT_XML_PATH='/sm/xml'"
val.32 = "Y"
val.33 = ""
val.34 = "7000"
val.35 = "Y"
val.36 = "Y"
val.37 = "Y"

rc = 4
i = 1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #createj2eeserver failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: createj2eeserver */
rc = CB390CFG("-action 'createj2eeserver' -xmlinput
'inputcreatej2eeserver.xml' -input 'tempin'
-output 'createj2eeserver'")
if (rc == 4) then do
  say "FCT Test #createj2eeserver failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("createj2eeserver"  "ALL")
if (rc == 4) then do
  say "FCT Test #createj2eeserver failed while XMLPARSE"
  exit
end
say "FCT Test #createj2eeserver completed"
return 0

exit
```

```
error:
say "Error in FCT Test #createj2eeserver" rc "at line" sigl
say sourceline(sigl)
exit
```

Action “deletej2eeserver”

This action causes a J2EE server to be deleted. This is a logical deletion. The deletion does not actually occur until the conversation this change is associated with is commited.

Syntax

```
►►──rc = CB390CFG──("── -action──'deletej2eeserver' ──────────────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬─────────────────────────────┬───────►
                                    └─ -input──'inputfilename' ──┘

►─ -output──'outputfilename' ──") ──────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for deletej2eeserver "inputdeletej2eeserver.xml" is listed in section “inputdeletej2eeserver.xml” on page 203. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputdeletej2eeserver.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputdeletej2eeserver.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this server action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputdeletej2eeserver.xml" on page 203.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the J2EE server	x

Example

Here is an example script:

```
/* REXX function */
/* Functiontest Test : deletej2eeserver*/

/* Dependencies: */
/* The conversation "API Functiontest" must be added*/
/* The J2EEserver "J2EESRV" must be added in the conversation
"API Functiontest"*/
call syscalls 'ON'
signal on error

say "FCT Test #deletej2eeserver"

name. = 0
name.1 = "conversationname"
name.2 = "j2eeservername"

val. = 0
val.1 = "API Functiontest"
val.2 = "J2EESRV"

rc = 4
i = 1

/*Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #deletej2eeserver failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: deletej2eeserver */
```

```
rc = CB390CFG("-action 'deletej2eeserver' -xmlinput
'inputdeletej2eeserver.xml' -input 'tempin'
-output 'deletej2eeserver'")
if (rc == 4) then do
  say "FCT Test #deletej2eeserver failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("deletej2eeserver"  "ALL")
if (rc == 4) then do
  say "FCT Test #deletej2eeserver failed while XMLPARSE"
  exit
end
say "FCT Test #deletej2eeserver completed"
exit

error:
say "Error in FCT Test #deletej2eeserver" rc "at line" sigl
say sourceline(sigl)
exit
```

Action “changej2eeserver”

This action causes attributes of the named J2EE server to be changed.

Syntax

```
►►──rc = CB390CFG──("── -action──'changej2eeserver' ──────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬──────────────────────────┬─────────────►
                                     └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")───────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for changej2eeserver "inputchangej2eeserver.xml" is listed in section “inputchangej2eeserver.xml” on page 204. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputchangej2eeserver.xml"`. Otherwise specify the complete

| location to the default xml file by setting this parameter to
| `"/usr/lpp/WebSphere/samples/smapi/inputchangej2eeserver.xml"`.
| If you want to use your own default xml file you **must** specify the
| complete directory of the file or you **must** set the
| `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file
The table below includes all of the attributes that are known for this server action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputchangej2eeserver.xml" on page 204.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the J2EE server	x
serverdescription	Description of the J2EE server	
identityofthecontrolregion		x
identityoftheserverregion		x
serverregionstacksize	Numerical	x
productionserver	Allowed values are "Y" or "N"	x
debuggerallowed	Allowed values are "Y" or "N"	x
isolationpolicy	Allowed values are `One_Transaction_Per_server_Region` `Multiple_Transactions_Per_Server_Region`	x
replicationpolicy	Allowed values are `One_Per_Server` `Replicate_As_Needed`	x
serverregionrequiresjvm	Allowed values are "Y" or "N"	x
serverregionjvmname	Name of the JVM	x
localidentity	Name of the local identity	x

Parameter name	Values	Required
remoteidentity	Name of the remote identity	x
transactionfactory	Allowed values are "Y" or "N"	x
allowserverregiongarbagecollection	Allowed values are "Y" or "N"	x
garbagecollectioninterval	Numerical value between 0 and 2G	x
logstreamname	Name of the logstream	x
procname		x
allownonathenticatedclients	Allowed values are "Y" or "N"	x
allowuseridpasswd	Allowed values are "Y" or "N"	x
useridpassticket	Allowed values are "Y" or "N"	x
usedce	Allowed values are "Y" or "N"	x

dcequalityofprotection	Allowed values are
	`No_Protection`
	`Integrity`
	`Confidentiality`
		x

dcekeytabfile	Name of the DCE Keytab File	x
allowssl	Allowed values are "Y" or "N"	x
sslracfkeyring		x
sslv2timeout	Numeric value between 1 and 100	x
sslv3timeout	Numeric value between 1 86400	x

security	Preference Of Security Type
	This specifies the sequence of the security. To specify security in the default xml file there must be set an element for each security type. If a security type is specified the attribute **must** be set.
	To specify in REXX only one element must be specified. To set the security type `security value [value value ...]` where `value` specifies the value of the Security type. If security is specified at least one value **must** be set. Allowed values are
	`ISM_DCE`
	`ISM_UserID_Password`
	`ISM_Pass_Ticket`
	`ISM_SSL`	

smfwrserveractivity	Allowed values are "Y" or "N"	x
smfwrcontaineractivity	Allowed values are "Y" or "N"	x
smfwrserverinterval	Allowed values are "Y" or "N"	x
smfwrcontainerinterval	Allowed values are "Y" or "N"	x

Parameter name	Values	Required
smfintervallength	Numeric value between 15 and 86400 or 0	x
environment	To specify the environment in the default xml there must be set an element for each environment type. If an environment type is specified the attributes **must** be set. To specify the environment in REXX only one element must be specified. To set the environment type `environment name ='value' [name ='value' ...]`, where `name` specifies the environment name and value the value of the environment. If the environment is specified at least one `name ='value'` pair **must** be specified.	
allowsslclientcerts	Allowed values are "Y" or "N"	x
olthostname	Character (256)	x
oltport	Character value between 1 and 65535	x
acceptassertedid	Allowed values are "Y" or "N"	x
allowkerberos	Allowed values are "Y" or "N"	x
sendassertedid	Allowed values are "Y" or "N"	x

For the J2EE server properties some changes have occured between the SM-EUI and the Scripting API. Below there is a table of the different values.

Parameter for "DCE Quality Of Protection"

Script value	GUI value
Integrity	Message Integrity
Confidentiality	Message Confidentiality

Example

Here is an example script:

```
/* REXX function */
/* Functiontest Test:  changej2eeserver*/
/* Dependencies: */
/* The conversation "API Functiontest" must be added*/
/* The J2EE server "J2EESRV" must be added in the conversation
"API Functiontest"*/
call syscalls 'ON'
signal on error

say "FCT Test #changej2eeserver"
```

```
name. = 0
name.1  = "conversationname"
name.2  = "j2eeservername"
name.3  = "identityofthecontrolregion"
name.4  = "identityoftheserverregion"
name.5  = "serverregionstacksize"
name.6  = "productionserver"
name.7  = "debuggerallowed"
name.8  = "isolationpolicy"
name.9  = "replicationpolicy"
name.10 = "serverregionrequiresjvm"
name.11 = "serverregionjvmname"
name.12 = "localidentity"
name.13 = "remoteidentity"
name.14 = "transactionfactory"
name.15 = "allowserverregiongarbagecollection"
name.16 = "garbagecollectioninterval"
name.17 = "logstreamname"
name.18 = "procname"
name.19 = "allownonauthenticatedclients"
name.20 = "allowuseridpasswd"
name.21 = "useridpassticket"
name.22 = "usedce"
name.23 = "dcequalityofprotection"
name.24 = "dcekeytabfile"
name.25 = "security"
name.26 = "allowssl"
name.27 = "j2eeserverdescription"
name.28 = "sslv2timeout"
name.29 = "sslv3timeout"
name.30 = "environment"
name.31 = "olthostname"
name.32 = "oltport"
name.33 = "allowsslclientcerts"
name.34 = "allowkerberos"
name.35 = "sendassertedid"
name.36 = "acceptassertedid"


val. = 0
val.1  = "API Functiontest"
val.2  = "J2EESRV"
val.3  = "IBMUSER"
val.4  = "IBMUSER"
val.5  = "0"
val.6  = "Y"
val.7  = "N"
val.8  = "Multiple_Transactions_Per_Server_Region"
val.9  = "One_Per_Server"
val.10 = "N"
val.11 = ""
val.12 = "CBGUEST"
val.13 = "CBGUEST"
val.14 = "N"
val.15 = "Y"
val.16 = "50000"
val.17 = ""
val.18 = "BBOASR1"
val.19 = "Y"
val.20 = "Y"
val.21 = "N"
val.22 = "N"
val.23 = "No_Protection"
val.24 = ""
val.25 = "ISM_UserID_Password ISM_DCE"
val.26 = "N"
val.27 = "J2EESRV Description modified"
val.28 = "200"
val.29 = "500"
val.30 = "CLASSPATH='testchange1' PATH='testchange2'"
val.31 = ""
val.32 = "9000"
val.33 = "N"
val.34 = "Y"
val.35 = "Y"
val.36 = "Y"


rc = 4
i = 1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #changej2eeserver failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: changeeserver */
rc = CB390CFG("-action 'changej2eeserver' -xmlinput
'inputchangej2eeserver.xml' -input 'tempin'
-output 'changej2eeserver'")
if (rc == 4) then do
  say "FCT Test #changej2eeserver failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("changej2eeserver"  "ALL")
if (rc == 4) then do
  say "FCT Test #changej2eeserver failed while XMLPARSE"
  exit
end
say "FCT Test #changej2eeserver completed"
return 0
exit
```

```
error:
say "Error in FCT Test #changej2eeserver" rc "at line" sigl
say sourceline(sigl)
exit
```

Action “listj2eeserver”

This action causes the named J2EE server to be listed. If the J2EE server name equals "*" then all J2EE servers will be listed.

Syntax

```
►►──rc = CB390CFG──("── -action──'listj2eeserver' ──────────────────────►

►── -xmlinput──'defaultxmlfilename' ──┬──────────────────────────┬──────►
                                      └─ -input──'inputfilename' ─┘

►── -output──'outputfilename' ──")─────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for listj2eeserver "inputlistj2eeserver.xml" is listed in section “inputlistj2eeserver.xml” on page 206. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory. If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputlistj2eeserver.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputlistj2eeserver.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
: This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
: This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this J2EE server action. The required ones must be defined in the default xml file or can be defined by the XMLGEN (“Chapter 5. XMLGEN” on page 169) script. The default xml file is listed in section “inputlistj2eeserver.xml” on page 206.

Parameter name	**Values**	**Required**
conversationname	Name of the conversation	x
servername	"*" to list all J2EE server in the conversation	x

Example

Here is an example script:

```
/* REXX function */
/* Functiontest Test : listj2eeserver*/

/* Dependencies: */
/* For Part 1: The conversation "API Functiontest" must be added and
   the J2EE server "J2EESRV" must be added in the conversation
   "API Functiontest"*/
/* For Part 2: The conversation "API Functiontest" must be added */
call syscalls 'ON'
signal on error

say "FCT Test #listj2eeserver A"

name. = 0
name.1 = "conversationname"
name.2 = "j2eeservername"

/* Part 1: */
/* List special J2EE server */
val. = 0
val.1 = "API Functiontest"
val.2 = "J2EESRV"

rc = 4
i = 1

/*Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #listj2eeserver failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: listj2eeserverA */
rc = CB390CFG("-action 'listj2eeserver' -xmlinput 'inputlistj2eeserver.xml'
-input 'tempin' -output 'listj2eeserverA'")
if (rc == 4) then do
  say "FCT Test #listj2eeserverA failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("listj2eeserverA"  "ALL")
if (rc == 4) then do
  say "FCT Test #listj2eeserverA failed while XMLPARSE"
  exit
end


/* Part 2: */
/* List all J2EE server */

say "FCT Test #listj2eeserver B"

val.2 = "*"

rc = 4
i = 1

/*Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #listj2eeserverb failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: listj2eeserverB */
rc = CB390CFG("-action 'listj2eeserver' -xmlinput 'inputlistj2eeserver.xml'
-input 'tempin' -output 'listj2eeserverB'")
if (rc == 4) then do
  say "FCT Test #listj2eeserverB failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("listj2eeserverB"  "ALL")
if (rc == 4) then do
  say "FCT Test #listj2eeserverB failed while XMLPARSE"
  exit
end
say "FCT Test #listj2eeserverB completed"
exit
```

```
error:
say "Error in FCT Test #listj2eeserver" rc "at line" sigl
say sourceline(sigl)
exit
```

J2EE application

These functions are for processing and listing of J2EE applications.

Syntax

```
►►──rc = CB390CFG──("── -action──'──┬─processearfile-──────────┬─'──────────►
                                    ├─listj2eeapplication-─────┤
                                    ├─deletej2eeapplication-───┤
                                    ├─listj2eemodules-─────────┤
                                    └─listj2eecomponent-───────┘

►─ -xmlinput──'defaultxmlfilename'──┬──────────────────────────┬───────────►
                                    └─ -input──'inputfilename'─┘

►─ -output──'outputfilename'──")───────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "4", an error has occurred while processing the action.

action

processearfile Processes a resolved ear file.

listj2eeapplication
Causes an application to be listed.

deletej2eeapplication
Causes an application to be deleted.

listj2eemodules Causes a module to be listed.

listj2eecomponent
Causes a component to be listed.

-xmlinput
This is the default xml file. In this file all required parameters for the action which should be performed **must** be specified. This file is a xml file with a document type definition (DTD). The DTD only specifies the structure of the document. The user can specify default values for each parameter. These parameters can be

overriden by the REXX script via the input parameter. All default xml files are listed in “Chapter 9. Default XML files” on page 185. The parameters in these files are set to the default values of the SM Administration EUI.

The default xml file **must** be present in the path that is specified by the environment variable `DEFAULT_CLIENT_XML_PATH` or the user has to specify its path.

Example: `-xmlinput 'inputprocessearfile.xml'` specifies the default input xml file in the `DEFAULT_CLIENT_XML_PATH`. But `-xmlinput './inputprocessearfile.xml'` specifies the file in the current directory.

The user can modify the default path for the default xml files by setting the environment variable `DEFAULT_CLIENT_XML_PATH` to another existing path. Make sure that the path exists and that the default xml file that will be used is in this directory.

-input
This is an optional parameter for the CB390CFG API. It specifies the input file which contains the name value pairs that should override the parameters of the default xml file. To generate a xml file by using REXX variables there is a tool called XMLGEN. This tool is described in “Chapter 5. XMLGEN” on page 169.

Important: The input file will be deleted after the parameters are merged with the default xml file.

-output
The output file contains further information. In the description of each J2EE application action there is an example output file. The general output format for a J2EE application looks like this:

```
administratorname.1 CBADMIN
conversationname.1 API Functiontest
j2eeapplicationname.1 testApp
j2eeservername.1 BBOASR4
sysplexname.1 PLEX1
status 0
message.1 OK
count 1
```

Action “processearfile”

This function is for processing an ear file. In order to do this, the ear file has to be saved via the SM EUI on the server side. The resolved ear file can then be used as input for the SM Scripting API.

Syntax

```
►►──rc = CB390CFG──("── -action──'processearfile' ──────────────────────►
►─ -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬──────►
                                    └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")───────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4", an error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for processearfile "inputprocessearfile.xml" is listed in section "inputprocessearfile.xml" on page 224. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

  If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory, you only need to type the filename `"inputprocessearfile.xml"`. Otherwise, specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputprocessearfile.xml"`. If you want to use your own default xml file, you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
: This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
: This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this server action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputprocessearfile.xml" on page 224.

Parameter name	Values	Required
conversationname	Name of the conversation	x
j2eeservername	Name of the J2EE server	x
earfilename	Name of the resolved ear file **Note:** It has to be the fully qualified name, e.g. `/tmp/testBean_rsolved.ear`	x

Example

Here is an example script:

```
call syscalls 'ON'
signal on error

say "FCT Test #processearfile"

name. = 0
name.1 = "conversationname"
name.2 = "j2eeservername"
name.3 = "earfilename"

val. = 0
val.1 = "API Functiontest"
val.2 = "BBOASR4"
val.3 = "/tmp/testApp_resolved.ear"


rc = 4
i = 1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #processearfile failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: importApplicationfamily */
rc = CB390CFG("-action 'processearfile' -xmlinput 'inputprocessearfile.xml'
-input 'tempin' -output 'processearfile'")
if (rc == 4) then do
  say "FCT Test #processearfile failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("processearfile"  "ALL")
if (rc == 4) then do
  say "FCT Test #processearfile failed while XMLPARSE"
  exit
```

```
end
say "FCT Test #processearfile completed"
return 0
exit

error:
say "Error in FCT Test #processearfile" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1  =  CBADMIN
conversationname.1  =  API Functiontest
j2eeapplicationname.1  =  testApp
j2eeservername.1  =  BBOASR4
sysplexname.1  =  PLEX1
status  =  0
message.1  =  OK
count  =  1
```

Action “listj2eeapplication”

This action causes the named application to be listed. If the application name equals "*" then all applications on the specified J2EE server will be listed.

Syntax

```
►►──rc = CB390CFG──("── -action──'listj2eeapplication' ─────────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬──────────────────────────┬────────────►
                                     └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")───────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for listj2eeapplication "inputlistj2eeapplication.xml" is listed in section “inputlistj2eeapplication.xml” on page 198. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename "`inputlistj2eeapplication.xml`". Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputlistj2eeapplication.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file
The table below includes all of the attributes that are known for this server action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputlistj2eeapplication.xml" on page 198.

Parameter name	Values	Required
conversationname	Name of the conversation	x
j2eeservername	Name of the J2EE server	x
j2eeapplicationname	Name of an application or "*" to list all installed applications on specied J2EE server.	x

Example
Here is an example script:

```
/* REXX function */
/* Functiontest Test : listj2eeapplication */

call syscalls 'ON'
signal on error

say "FCT Test #listj2eeapplication"

name. = 0
name.1 = "conversationname"
name.2 = "j2eeservername"
name.3 = "j2eeapplicationname"

/* Part 1: */
/* List special J2EE application  */
```

```
val. = 0
val.1 = "API Functiontest"
val.2 = "BBOASR4"
val.3 = "testApp"

rc = 4
i = 1
say "TEST listj2eeapplication A"

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #listj2eeapplication A failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: listj2eeapplication */
rc = CB390CFG("-action 'listj2eeapplication' -xmlinput
'inputlistj2eeapplication.xml' -input 'tempin' -output
'listj2eeapplicationA'")
if (rc == 4) then do
  say "FCT Test #listj2eeapplication A failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("listj2eeapplicationA"  "ALL")
if (rc == 4) then do
  say "FCT Test #listj2eeapplication A failed while XMLPARSE"
  exit
end


/* Part 2: */
/* List all J2EE application  */
say "TEST listj2eeapplication B"

val. = 0
val.1 = "API Functiontest"
val.2 = "BBOASR4"
val.3 = "*"

rc = 4
i = 1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #listj2eeapplication B failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: listj2eeapplication */
rc = CB390CFG("-action 'listj2eeapplication' -xmlinput
'inputlistj2eeapplication.xml' -input 'tempin' -output
'listj2eeapplicationB'")
if (rc == 4) then do
  say "FCT Test #listj2eeapplication B failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("listj2eeapplicationB"  "ALL")
if (rc == 4) then do
  say "FCT Test #listj2eeapplication B failed while XMLPARSE"
  exit
end
say "FCT Test #listj2eeapplication B completed"
exit

error:
say "Error in FCT Test #listj2eeapplication" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
conversationname.1 API Functiontest
j2eeapplicationname.1 testApp
j2eeservername.1 BBOASR4
sysplexname.1 PLEX1
status 0
message.1 OK
count 1
```

Action “deletej2eeapplication”

This action causes the named J2EE application to be deleted. This is a logical deletion. The deletion does not occur until the conversation this change is associated with is commited.

Syntax

```
►►──rc = CB390CFG──("── -action──'deletej2eeapplication' ─────────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬──────────────────────────┬─────────────►
                                     └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for deletej2eeapplication "inputdeletej2eeapplication.xml" is listed in section "inputdeletej2eeapplication.xml" on page 199. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

  If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputdeletej2eeapplication.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputdeletej2eeapplication.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
: This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
: This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputdeletej2eeapplication.xml" on page 199.

Parameter name	Values	Required
conversationname	Name of the conversation	x
j2eeservername	Name of the J2EE server	x
j2eeapplicationname	Name of the application to be deleted	x

Example

Here is an example script:

```
/* REXX function */
/* Functiontest Test : listj2eeapplication */

call syscalls 'ON'
signal on error

say "FCT Test #listj2eeapplication"

name. = 0
name.1 = "conversationname"
name.2 = "j2eeservername"
name.3 = "j2eeapplicationname"

/* Part 1: */
/* List special J2EE application  */
val. = 0
val.1 = "API Functiontest"
val.2 = "BBOASR4"
val.3 = "testApp"

rc = 4
i = 1
say "TEST listj2eeapplication A"

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #listj2eeapplication A failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: listj2eeapplication */
rc = CB390CFG("-action 'listj2eeapplication' -xmlinput
'inputlistj2eeapplication.xml' -input 'tempin' -output
'listj2eeapplicationA'")
if (rc == 4) then do
  say "FCT Test #listj2eeapplication A failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("listj2eeapplicationA"  "ALL")
if (rc == 4) then do
  say "FCT Test #listj2eeapplication A failed while XMLPARSE"
  exit
end


/* Part 2: */
/* List all J2EE application  */
say "TEST listj2eeapplication B"

val. = 0
val.1 = "API Functiontest"
val.2 = "BBOASR4"
val.3 = "*"

rc = 4
i = 1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #listj2eeapplication B failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: listj2eeapplication */
rc = CB390CFG("-action 'listj2eeapplication' -xmlinput
'inputlistj2eeapplication.xml' -input 'tempin' -output
'listj2eeapplicationB'")
if (rc == 4) then do
  say "FCT Test #listj2eeapplication B failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("listj2eeapplicationB"  "ALL")
if (rc == 4) then do
  say "FCT Test #listj2eeapplication B failed while XMLPARSE"
  exit
end
say "FCT Test #listj2eeapplication B completed"
exit

error:
say "Error in FCT Test #listj2eeapplication" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
conversationname.1 API Functiontest
j2eeapplicationname.1 testApp
j2eeservername.1 BBOASR4
sysplexname.1 PLEX1
status 0
message.1 OK
count 1
```

Action “listj2eemodules”

This action causes the named modules to be listed. If the module’s name equals "*", then all modules on the specified J2EE application will be listed.

Syntax

```
►►──rc = CB390CFG──("── -action──'listj2eemodules' ──────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬─────────────────────────────┬───────────►
                                     └─ -input──'inputfilename' ──┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for listj2eemodules "inputlistj2eemodules.xml" is listed in section “inputlistj2eemodules.xml” on page 200. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

  If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputlistj2eemodules.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/listj2eemodules.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
: This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
: This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this action. The *required* ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputlistj2eemodules.xml" on page 200.

Parameter name	**Values**	**Required**
conversationname	Name of the conversation	x
j2eeservername	Name of the J2EE server	x
j2eeapplicationname	Name of a J2EE application	x
j2eemodulename	Name of a module or "*" to list all modules associated with the specified J2EE application.	x

Example

Here is an example script:

```
Here is an example script:

call syscalls 'ON'
signal on error

say "FCT Test #listj2eeapplication"

name. = 0
name.1 = "conversationname"
name.2 = "j2eeservername"
name.3 = "j2eeapplicationname"

/* Part 1: */
/* List special J2EE application  */
val. = 0
val.1 = "API Functiontest"
val.2 = "BBOASR4"
val.3 = "testApp"

rc = 4
i = 1
say "TEST listj2eeapplication A"

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #listj2eeapplication A failed while XMLGEN"
    exit
  end
  i = i+1
end;
```

```
/* Call the function: listj2eeapplication */
rc = CB390CFG("-action 'listj2eeapplication' -xmlinput
'inputlistj2eeapplication.xml' -input 'tempin' -output
'listj2eeapplicationA'")
if (rc == 4) then do
  say "FCT Test #listj2eeapplication A failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("listj2eeapplicationA"  "ALL")
if (rc == 4) then do
  say "FCT Test #listj2eeapplication A failed while XMLPARSE"
  exit
end


/* Part 2: */
/* List all J2EE application  */
say "TEST listj2eeapplication B"

val. = 0
val.1 = "API Functiontest"
val.2 = "BBOASR4"
val.3 = "*"

rc = 4
i = 1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #listj2eeapplication B failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: listj2eeapplication */
rc = CB390CFG("-action 'listj2eeapplication' -xmlinput
'inputlistj2eeapplication.xml' -input 'tempin' -output
'listj2eeapplicationB'")
if (rc == 4) then do
  say "FCT Test #listj2eeapplication B failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("listj2eeapplicationB"  "ALL")
if (rc == 4) then do
  say "FCT Test #listj2eeapplication B failed while XMLPARSE"
  exit
end
say "FCT Test #listj2eeapplication B completed"
```

```
exit

error:
say "Error in FCT Test #listj2eeapplication" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
conversationname.1 API Functiontest
j2eeapplicationname.1 testApp
j2eeservername.1 BBOASR4
sysplexname.1 PLEX1
status 0
message.1 OK
count 1
```

Action “listj2eecomponents”

This action causes the named components to be listed. If the components name equals "*" then all components on the specified J2EE module will be listed.

Syntax

```
►►──rc = CB390CFG──(″── -action──’listj2eecomponents’ ──────────────────────►

►─ -xmlinput──’defaultxmlfilename’ ──┬──────────────────────────┬──────────►
                                     └─ -input──’inputfilename’ ─┘

►─ -output──’outputfilename’ ──″)──────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for listj2eecomponents "inputlistj2eecomponents.xml" is listed in section “inputlistj2eecomponents.xml” on page 199. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory. If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename "`listj2eecomponents.xml`". Otherwise specify the complete location to the default xml file by setting this parameter to "`/usr/lpp/WebSphere/samples/smapi/listj2eecomponents.xml`". If

| you want to use your own default xml file you **must** specify the
| complete directory of the file or you **must** set the
| `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file
The table below includes all of the attributes that are known for this action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputlistj2eecomponents.xml" on page 199.

Parameter name	**Values**	**Required**
conversationname	Name of the conversation	x
j2eeservername	Name of the J2EE server	x
j2eeapplicationname	Name of an application	x
j2eemodulename	Name of a module	x
j2eecomponentsname	Name of a component or "*" to list all components associated with the specified J2EE module.	x

Example
Here is an example script:

```
/* REXX function */
/* Functiontest Test : listj2eemodules */

/* Dependencies: */
/* Part 1: The conversation "API Functiontest" must be added,
   the server "APIFCT" must be added to the "API Functiontest" conversation
   and the application family "API_Funcitontest_Application" must be added */
/* Part 2: The conversation "API Functiontest" must be added and
   the server "APIFCT" must be added to the "API Functiontest" conversation*/
call syscalls 'ON'
signal on error

say "FCT Test #listj2eemodules special case"
```

```
name. = 0
name.1 = "conversationname"
name.2 = "j2eeservername"
name.3 = "j2eeapplicationname"
name.4 = "modulename"

/* Part 1: */
/* List special J2EE modules  */
val. = 0
val.1 = "API Functiontest"
val.2 = "BBOASR4"
val.3 = "testApp"
val.4 = "testBean_deploy.jar"


rc = 4
i = 1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #listj2eemodules A failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: listj2eemodules */
rc = CB390CFG("-action 'listj2eemodules' -xmlinput 'inputlistj2eemodules.xml'
-input 'tempin' -output 'listj2eemodules'")
if (rc == 4) then do
  say "FCT Test #listj2eemodules failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("listj2eemodules"  "ALL")
if (rc == 4) then do
  say "FCT Test #listj2eemodules failed while XMLPARSE"
  exit
end

say "TEST listj2eemodules all cases"

/* Part 2: */
/* List all J2EE modules  */

val. = 0
val.1 = "API Functiontest"
val.2 = "BBOASR4"
val.3 = "PolicyIVP"
val.4 = "*"

rc = 4
```

```
i = 1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #listj2eemodules B failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: listj2eemodules */
rc = CB390CFG("-action 'listj2eemodules' -xmlinput 'inputlistj2eemodules.xml'
-input 'tempin' -output 'listj2eemodulesB'")
if (rc == 4) then do
  say "FCT Test #listj2eemodules B failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("listj2eemodulesB"  "ALL")
if (rc == 4) then do
  say "FCT Test #listj2eemodules B failed while XMLPARSE"
  exit
end

exit

error:
say "Error in FCT Test #listj2eemodules" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
conversationname.1 API Functiontest
j2eeapplicationname.1 testApp
j2eeservername.1 BBOASR4
modulename.1 testBean_deploy.jar
sysplexname.1 PLEX1
status 0
message.1 OK
count 1
```

Server Instances

These functions are for the modifications of a server instance.

Syntax

```
►►──rc = CB390CFG──("── -action──'──┬─create- ─┬──serverinstance──'──────────►
                                    ├─delete- ─┤
                                    ├─change- ─┤
                                    └─list- ───┘

►── -xmlinput──'defaultxmlfilename' ──┬──────────────────────────────┬────────►
                                      └─ -input──'inputfilename' ────┘

►── -output──'outputfilename' ──")───────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "4" an error has occurred while processing the action.

-action

createserverinstance
Causes a new server instance to be created.

deleteserverinstance
Causes the server instance to be deleted.

changeserverinstance
Causes the server instance to be changed.

listserverinstance
Causes the server instance to be listed

-xmlinput
This is the default xml file. In this file all required parameters for the action which should be performed must be specified. This file is a xml file with a document type definition (DTD). The DTD only specifies the structure of the document. The user can specify default values for each parameter. These parameters can be overriden by the REXX script via the input parameter. All default xml files are listed in "Chapter 9. Default XML files" on page 185. The parameters in these files are set to the default values of the SM Administration EUI.

The default xml file must be present in the path that is specified by the environment variable `DEFAULT_CLIENT_XML_PATH` or the user has to specify its path.

Example: `-xmlinput 'inputcreateserverinstance.xml'` specifies the default input xml file in the `DEFAULT_CLIENT_XML_PATH`.
But `-xmlinput './inputcreateserverinstance.xml'` specifies the file in the current directory.

The user can modify the default path for the default xml files by setting the environment variable `DEFAULT_CLIENT_XML_PATH` to another existing path. Make sure that the path exists and that the default xml file that will be used is in this directory.

-input
This is an optional parameter for the CB390CFG API. It specifies the input file which contains the name value pairs that should override the parameters of the default xml file. To generate a xml file by using REXX variables there is a tool called XMLGEN. This tool is described in "Chapter 5. XMLGEN" on page 169.

Important: The input file will be deleted after the parameters are merged with the default xml file.

output
The output file contains further information. In the description of each serverinstance action there is an example output file. The general output format for a serverinstance action looks like this:

```
administratorname.1 AdministratorName
configportnumber.1  ConfiguredPortNumber
conversationname.1 ConversationName
logstreamname.1 LogstreamName
serverinstancedescription.1 ServerInstanceDescription
serverinstancename.1 ServerInstanceName
servername.1 ServerName
sslfirewallport.1 SSLFirewallPortNumber
sysplexname.1 SysplexName
systemname.1 SystemName
status 0|4
message.1 OK|ErrorMessage
count NumberOfListedServerInstances
```

Action "createserverinstance"

This action causes a new server instance to be created.

Syntax

```
►►──rc = CB390CFG──("── -action──'createserverinstance' ─────────────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬─────────────►
                                    └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename

This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for createserverinstance "inputcreateserverinstance. xml" is listed in section "inputcreateserverinstance.xml" on page 207. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputcreateserverinstance.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputcreateserverinstance.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename

This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename

This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this serverinstance action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputcreateserverinstance.xml" on page 207.

Parameter name	Values	Required
conversationname	Name of the conversationr	x
servername	Name of the server	x
serverinstancename	Name of the server instance	x
serverinstancedescription	Description of the server instance	x
systemname	Name of the system	x
logstreamname	Name of the logstream	x

Parameter name	Values	Required
environment	To specify the environment in the default xml there must be set an element for each environment type. If an environment type is specified the attributes **must** be set. To specify the environment in REXX only one element must be specified. To set the environment type `environment name ='value' [name ='value' ...]` , where `name` specifies the environment name and value the value of the environment. If the environment is specified at least one `name ='value'` pair **must** be specified.	
configportnumberr	Numeric value between 0 and 65535	x
sslfirewallport	Numeric value between 0 and 65535	x

Example

Here is an example script:

```
/* REXX function */
call syscalls 'ON'
signal on error

say "FCT Test #11"

name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "serverinstancename"
name.4 = "serverinstancedescription"
name.5 = "systemname"
name.6 = "logstreamname"
name.7 = "environment"
name.8 = "configportnumber"
name.9 = "sslfirewallport"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"
val.3 = "APIFCTSI"
val.4 = "API Functiontest ServerInstance"
val.5 = "SY1"
val.6 = ""
val.7 = "CLASSPATH='test1' PATH='test2'"
val.8 = "12345"
val.9 = "9000"
```

```
rc=4
i=1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #11 failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: createserverinstance */
rc = CB390CFG("-action 'createserverinstance'
 -xmlinput 'inputcreateserverinstance.xml'
 -input 'tempin' -output 'FCT11'")
if (rc == 4) then do
  say "FCT Test #11 failed"
  exit
end

/* Parse the result */
rc = XMLPARSE("FCT11"  "ALL")
if (rc == 4) then do
  say "FCT Test #11 failed while XMLPARSE"
  exit
end
say "FCT Test #11 completed"
return 0
exit

error:
say "Error in FCT Test #11" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
configportnumber.1 12345
conversationname.1 API Functiontest
logstreamname.1
serverinstancedescription.1 API Functiontest ServerInstance
serverinstancename.1 APIFCTSI
servername.1 APIFCT
sslfirewallport.1 9000
sysplexname.1 PLEX1
systemname.1 SY1
status 0
message.1 OK
count 1
```

Action “deleteserverinstance”

This action causes the named server instance to be deleted. This is a logical deletion. The deletion does not occur until the conversation this change is associated with is commited.

Syntax

```
►►──rc = CB390CFG──("── -action──'deleteserverinstance' ──────────────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬──────────────►
                                    └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")─────────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for deleteserverinstance "inputdeleteserverinstance. xml" is listed in section “inputdeleteserverinstance.xml” on page 208. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputdeleteserverinstance.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputdeleteserverinstance.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file
The table below includes all of the attributes that are known for this serverinstance action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputdeleteserverinstance.xml" on page 208.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	x
serverinstancename	Name of the server instance	x

Example
Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name.1 = "conversationname"
name.2 = "servername"
name.3 = "serverinstancename"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"
val.3 = "APIFCTSI"

rc = 4
i = 1

do while(val.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #13 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'deleteserverinstance'
               -xmlinput 'inputdeleteserverinstance.xml'
               -input 'tempin' -output 'FCT13'")
if (rc == 4) then do
  say "FCT Test #13 failed"
  exit
end
```

```
exit

error:
say "Error in FCT Test #13" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
conversationname.1 API Functiontest
logstreamname.1
serverinstancedescription.1 API Functiontest Serverinstance Description modified
serverinstancename.1 APIFCTSI
servername.1 APIFCT
sysplexname.1 PLEX1
systemname.1 SY1
environment.1 CLASSPATH = 'testchange1' PATH = 'testchange2'
status 0
message.1 OK
count 1
```

Action “changeserverinstance”

This action causes the attributes of the named server instance to be changed.

Syntax

```
►►──rc = CB390CFG──("── -action──'changeserverinstance' ──────────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬─────────────────────────────┬─────────►
                                     └─ -input──'inputfilename' ──┘

►─ -output──'outputfilename' ──")────────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for changeserverinstance "inputchangeserverinstance. xml" is listed in “inputchangeserverinstance.xml” on page 208. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

  If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename "`inputchangeserverinstance.xml`". Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputchangeserverinstance.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file must **contain** all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file
The table below includes all of the attributes that are known for this serverinstance action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputchangeserverinstance.xml" on page 208.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	x
serverinstancename	Name of the server instance	x
serverinstancedesription	Description of the server instance	x
systemname	Name of the system	x
logstreamname	Name of the logstream	x
environment	To specify the environment in the default xml there **must** be set an element for each environment type. If an environment type is specified the attributes must be set. To specify the environment in REXX only one element must be specified. To set the environment type `environment name ='value' [name ='value' ...]` , where `name` specifies the environment name and value the value of the environment. If the environment is specified atleast one `name ='value'` pair **must** be specified.	
configportnumberr	Numeric value between 0 and 65535	x

Parameter name	Values	Required
sslfirewallport	Numeric value between 0 and 65535	x

Example

Here is an example script:

```
/* REXX function */
call syscalls 'ON'
signal on error

say "FCT Test #12"

name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "serverinstancename"
name.4 = "serverinstancedescription"
name.5 = "systemname"
name.6 = "logstreamname"
name.7 = "environment"
name.8 = "configportnumber"
name.9 = "sslfirewallport"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"
val.3 = "APIFCTSI"
val.4 = "API Functiontest Serverinstance Description modified"
val.5 = "SY1"
val.6 = ""
val.7 = "CLASSPATH='testchange1' PATH='testchange2'"
val.8 = "50000"
val.9 = "11000"

rc = 4
i = 1

/* Generate XML Input */
do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #12 failed while XMLGEN"
    exit
  end
  i = i+1
end;

/* Call the function: changeserverinstance */
rc = CB390CFG("-action 'changeserverinstance'
 -xmlinput 'inputchangeserverinstance.xml'
 -input 'tempin' -output 'FCT12'")
if (rc == 4) then do
  say "FCT Test #12 failed"
  exit
```

```
end

/* Parse the result */
rc = XMLPARSE("FCT12"  "ALL")
if (rc == 4) then do
  say "FCT Test #12 failed while XMLPARSE"
  exit
end
say "FCT Test #12 completed"
exit

error:
say "Error in FCT Test #12" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
configportnumber.1 50000
conversationname.1 API Functiontest
logstreamname.1
serverinstancedescription.1 API Functiontest Serverinstance Description modifie
serverinstancename.1 APIFCTSI
servername.1 APIFCT
sslfirewallport.1 11000
sysplexname.1 PLEX1
systemname.1 SY1
status 0
message.1 OK
count 1
```

Action “listserverinstance”

This action causes the named server instance to be listed. If the server instance name equals "*" then all server instances will be listed.

Syntax

```
►►──rc = CB390CFG──("── -action──'listserverinstance' ─────────────────────►

►── -xmlinput──'defaultxmlfilename' ──┬───────────────────────────┬────────►
                                      └─ -input──'inputfilename' ─┘

►── -output──'outputfilename' ──")─────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "4" an error has occurred while processing the action.

defaultxmlfilename

This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for listserverinstance "inputlistserverinstance.xml" is listed in section “inputlistserverinstance.xml” on page 210. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputlistserverinstance.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputlistserverinstance.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename

This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename

This parameter specifies the name of the output file. It will be written into the "/tmp" directory.

Values of default xml file

The table below includes all of the attributes that are known for this serverinstance action. The required ones must be defined in the default xml file or can be defined by the XMLGEN (“Chapter 5. XMLGEN” on page 169) script. The default xml file is listed in section “inputlistserverinstance.xml” on page 210.

Parameter name	**Values**	**Required**
conversationname	Name of the conversation	x
servername	Name of the server	x
serverinstancename	Name of the server instance "*" to list all server instances pertaining to the server	x

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "serverinstancename"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"
val.3 = "APIFCTSI"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #14 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'listserverinstance'
               -xmlinput 'inputlistserverinstance.xml'
               -input 'tempin' -output 'FCT14'")
if (rc == 4) then do
  say "FCT Test #14 failed"
  exit
end
exit

error:
say "Error in FCT Test #14" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
conversationname.1 API Functiontest
logstreamname.1
serverinstancedescription.1 API Functiontest Serverinstance Description modifie
serverinstancename.1 APIFCTSI
servername.1 APIFCT
sysplexname.1 PLEX1
systemname.1 SY1
environment.1 CLASSPATH = 'testchange1' PATH = 'testchange2'
status 0
message.1 OK
count 1
```

Container

These functions are for the modifications of a container.

Syntax

```
►►──rc = CB390CFG──("── -action──'──┬─create- ─┬──container──'──────────────►
                                    ├─delete- ─┤
                                    ├─change- ─┤
                                    └─list- ───┘

►─ -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬────────────►
                                    └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")───────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "4" an error has occurred while processing the action.

-action

createcontainer	Causes a new container to be created.
deletecontainer	Causes the container to be deleted.
changecontainer	Causes the container to be changed.
listcontainer	Causes the container to be listed.

-xmlinput
This is the default xml file. In this file all required parameters for the action which should be performed must be specified. This file is a xml file with a document type definition (DTD). The DTD only specifies the structure of the document. The user can specify default values for each parameter. These parameters can be overriden by the REXX script via the input parameter. All default xml files are listed in “Chapter 9. Default XML files” on page 185. The parameters in these files are set to the default values of the SM Administration EUI.

The default xml file must be present in the path that is specified by the environment variable `DEFAULT_CLIENT_XML_PATH` or the user has to specify its path.

Example: `-xmlinput 'inputcreatecontainer.xml'` specifies the default input xml file in the `DEFAULT_CLIENT_XML_PATH`. But `-xmlinput './inputcreatecontainer.xml'` specifies the file in the current directory.

The user can modify the default path for the default xml files by setting the environment variable `DEFAULT_CLIENT_XML_PATH` to another existing path. Make sure that the path exists and that the default xml file that will be used is in this directory.

-input
This is an optional parameter for the CB390CFG API. It specifies the input file which contains the name value pairs that should override the parameters of the default xml file. To generate a xml file by using REXX variables there is a tool called XMLGEN. This tool is described in "Chapter 5. XMLGEN" on page 169.

Important: The input file will be deleted after the parameters are merged with the default xml file.

-output
The output file contains further information. In the description of each container action there is an example output file. The general output format for a container action looks like this:

```
aclcheckrequired.1 Y|N
activationisolation_policy.1 ActivationIsolationPolicyState
administratorname.1 AdministratorName
containerdescription.1 ContainerDescription
containername.1 ContainerName
conversationname.1 ConversationName
managedobjectrefresh_policy.1 ManagedobjectRefreshPolicyState
passivationconstraints.1 PassivationConstraintsState
servername.1 ServerName
sysplexname.1 SysplexName
transactionpolicy.1 TransactionPolicyState
status 0|4
message.1 OK|ErrorMessage
count NumberOfListedContainer
```

Action "createcontainer"

This action causes a new container to be created.

Syntax

```
►►──rc = CB390CFG──("── -action──'createcontainer' ──────────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬──────────────────────────┬────────►
                                     └─ -input──'inputfilename' ─┘
►─ -output──'outputfilename' ──")──────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for createcontainer "inputcreatecontainer.xml" is listed in section "inputcreatecontainer.xml" on page 210. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputcreatecontainer.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputcreatecontainer.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file
The table below includes all of the attributes that are known for this container action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputcreatecontainer.xml" on page 210.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	x
containername	Name of the container	x

Parameter name	Values	Required
containerdescription	Description of the container	x
aclcheckrequired	Allowed values are "Y"or "N"	x

activationisolationpolicy	Allowed values are
	`Transaction_Level`
	`Container_Level`
		x

passivationconstraints	Allowed values are
	`Pinned`
	`Pinned_For_Life_Of_Transaction`
	`Not Pinned`
		x

managedobjectrefreshpolicy	Allowed values are
	`At_Transaction_Recognition`
	`At_Activation`
		x

transactionpolicy	Allowed values are
	`Tx_Required`
	`Tx_MOFW_Isolated_HG`
	`Tx_MOFW_ Merged_ HG`
	`Tx_MOFW_Supports_Merged_HG`
		x

For the container properties some changes have occured between the SM-EUI and the Script-ing API. Below there are tables of the different values.

Parameter for "Passivation Constraints"

Script value	GUI value
Pinned_For_Life_Of Transaction	Pinned For Transaction Life

Parameter for "Managed Object Refresh Policy"

Script value	GUI value
At_Transaction_Recognition	Per Transaction

Parameter for "Transaction Policy"

Script value	GUI value
Tx_Required	Required
Tx_MOFW_Isolated_HG	Same Server Hybrid Global
Tx_MOFW_Merged_HG	Hybrid Global
Tx_MOFW_Supports_Merged_HG	Supports Same-Server Hybrid Global

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "containername"
name.4 = "containerdescription"
name.5 = "aclcheckrequired"
name.6 = "activationisolation_policy"
name.7 = "passivationconstraints"
name.8 = "managedobjectrefresh_policy"
name.9 = "transactionpolicy"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"
val.3 = "API_Functiontest_Container"
val.4 = "API Functiontest Container Description"
val.5 = "N"
val.6 = "Transaction_Level"
val.7 = "Not_Pinned"
val.8 = "At_Activation"
val.9 = "TX_Required"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #15 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'createcontainer' -xmlinput 'inputcreatecontainer.xml'
               -input 'tempin' -output 'FCT15'")
if (rc == 4) then do
  say "FCT Test #15 failed"
  exit
end
exit

error:
say "Error in FCT Test #15" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
aclcheckrequired.1 N
activationisolationpolicy.1 Transaction_Level
administratorname.1 CBADMIN
containerdescription.1 API Functiontest Container Description
containername.1 API_Functiontest_Container
conversationname.1 API Functiontest
managedobjectrefreshpolicy.1 At_Activation
passivationconstraints.1 Not_Pinned
servername.1 APIFCT
sysplexname.1 PLEX1
transactionpolicy.1 Tx_Required
status 0
message.1 OK
count 1
```

Action “deletecontainer”

This action causes the named container to be deleted. This is a logical deletion. The deletion does not occur until the conversation this change is associated with is commited.

Syntax

```
►►──rc = CB390CFG──("── -action──'deletecontainer' ──────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬─────────────►
                                    └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")─────────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for deletecontainer "inputdeletecontainer.xml" is listed in section “inputdeletecontainer.xml” on page 211. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

  If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputdeletecontainer.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputdeletecontainer.xml"`. If you want to use your own default xml file you **must** specify the

| complete directory of the file or you **must** set the
| `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file

The table below includes all of the attributes that are known for this container action. The required ones must be defined in the default xml file or can be defined by the XMLGEN (“Chapter 5. XMLGEN” on page 169) script. The default xml file is listed in section “inputdeletecontainer.xml” on page 211.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	x
containername	Name of the container	x

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "containername"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"
val.3 = "API_Functiontest_Container"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
```

```
    say "FCT Test #17 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'deletecontainer' -xmlinput
'inputdeletecontainer.xml' -input 'tempin' -output 'FCT17'")
if (rc == 4) then do
  say "FCT Test #17 failed"
  exit
end
exit

error:
say "Error in FCT Test #17" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
aclcheckrequired.1 Y
activationisolationpolicy.1 Container_Level
administratorname.1 CBADMIN
containerdescription.1 API Functiontest Container Description
containername.1 API_Functiontest_Container
conversationname.1 API Functiontest
managedobjectrefreshpolicy.1 At_Transaction_Recognition
passivationconstraints.1 Pinned
servername.1 APIFCT
sysplexname.1 PLEX1
transactionpolicy.1 Tx_MOFW_Supports_Merged_HG
status 0
message.1 OK
count 1
```

Action “changecontainer”

This action causes the attributes of the named container to be changed.

Syntax

```
►►──rc = CB390CFG──("── -action──'changecontainer' ──────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬────────────►
                                    └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for changecontainer "inputchangecontainer.xml" is listed in section "inputchangecontainer.xml" on page 212. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputchangecontainer.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputchangecontainer.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file
The table below includes all of the attributes that are known for this container action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputchangecontainer.xml" on page 212.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	x
containername	Name of the container	x
containerdescription	Description of the container	
aclcheckrequired	Allowed values are "Y"or "N"	x

Parameter name	Values	Required

activationisolationpolicy	Allowed values are
	`Transaction_Level`
	`Container_Level`
		x

passivationconstraints	Allowed values are
	`Pinned`
	`Pinned_For_Life_Of_Transaction`
	`Not_Pinned`
		x

managedobjectrefreshpolicy	Allowed values are
	`At_Transaction_Recognition`
	`At_Activation`
		x

transactionpolicy	Allowed values are
	`Tx_Required`
	`Tx_MOFW_Isolated HG`
	`Tx_MOFW_Merged_HG`
	`Tx_MOFW_Supports Merged_HG`
		x

For the container properties some changes have occured between the SM-EUI and the Script-ing API. Below there are tables of the different values.

Parameter for "Passivation Constraints"

Script value	GUI value
Pinned_For_Life_Of_Transaction	Pinned_For_Transaction_Life

Parameter for "Managed Object Refresh Policy"

Parameter name	Values
At_Transaction_Recognition	Per Transaction

Parameter for "Transaction Policy"

Script value	GUI value
Tx_Required	Required
Tx_MOFW_Isolated_HG	Same Server Hybrid Global
Tx_MOFW_Merged_HG	Hybrid Global
Tx_MOFW_Supports_Merged_HG	Supports Same-Server Hybrid Global

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "containername"
name.4 = "containerdescription"
name.5 = "aclcheckrequired"
name.6 = "activationisolation_policy"
name.7 = "passivationconstraints"
name.8 = "managedobjectrefresh_policy"
name.9 = "transactionpolicy"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"
val.3 = "API_Functiontest_Container"
val.4 = "API Functiontest Container Description"
val.5 = "Y"
val.6 = "Container_Level"
val.7 = "Pinned"
val.8 = "At_Transaction_Recognition"
val.9 = "TX_MOFW_Supports_Merged_HG"

rc = 4
i = 1

do while(name.i <>'0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #16 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'changecontainer' -xmlinput 'inputchangecontainer.xml'
               -input 'tempin' -output 'FCT16'")
if (rc == 4) then do
  say "FCT Test #16 failed"
  exit
end
exit

error:
say "Error in FCT Test #16" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
aclcheckrequired.1 Y
activationisolationpolicy.1 Container_Level
administratorname.1 CBADMIN
containerdescription.1 API Functiontest Container Description
containername.1 API_Functiontest_Container
conversationname.1 API Functiontest
managedobjectrefreshpolicy.1 At_Transaction_Recognition
passivationconstraints.1 Pinned
servername.1 APIFCT
sysplexname.1 PLEX1
transactionpolicy.1 Tx_MOFW_Supports_Merged_HG
status 0
message.1 OK
count 1
```

Action “listcontainer”

This action causes the named container to be listed. If the container name equals "*"then all container will be listed.

Syntax

```
►►──rc = CB390CFG──("── -action──'listcontainer' ─────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬──────────────────────────┬──────────►
                                     └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for listcontainer "inputlistcontainer.xml" is listed in section “inputlistcontainer.xml” on page 213. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputlistcontainer.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputlistcontainer.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory. Values of default xml file The table below includes all of the attributes that are known for this container action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputlistcontainer.xml" on page 213.

Values of default xml file

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	x
containername	Name of the container "*" to list all container pertaining to the specified server	x

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "containername"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"
val.3 = "API_Functiontest_Container"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
```

```
    say "FCT Test #18 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'listcontainer' -xmlinput 'inputlistcontainer.xml'
               -input 'tempin' -output 'FCT18'")
if (rc == 4) then do
  say "FCT Test #18 failed"
  exit
end
exit

error:
say "Error in FCT Test #18" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
aclcheckrequired.1 Y
activationisolationpolicy.1 Container_Level
administratorname.1 CBADMIN
containerdescription.1 API Functiontest Container Description
containername.1 API_Functiontest_Container
conversationname.1 API Functiontest
managedobjectrefreshpolicy.1 At_Transaction_Recognition
passivationconstraints.1 Pinned
servername.1 APIFCT
sysplexname.1 PLEX1
transactionpolicy.1 Tx_MOFW_Supports_Merged_HG
status 0
message.1 OK
count 1
```

LRM

These functions are for the modifications of a LRM.

Syntax

```
►►──rc = CB390CFG──("── -action──’──┬─create-──┬──lrm──’──────────────────►
                                    ├─delete-──┤
                                    ├─change-──┤
                                    └─list-────┘

►── -xmlinput──’defaultxmlfilename’──┬────────────────────────────┬───────►
                                     └─ -input──’inputfilename’ ──┘
►─ -output─'outputfilename' ─")───────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "0" if error has occurred while processing the action.

-action

createlrm	Causes a new lrm to be created.
deletelrm	Causes the lrm to be deleted.
changelrm	Causes the lrm to be changed.
listlrm	Causes the lrm to be listed.

-xmlinput
This is the default xml file. In this file all required parameters for the action which should be performed must be specified. This file is a xml file with a document type definition (DTD). The DTD only specifies the structure of the document. The user can specify default values for each parameter. These parameters can be overriden by the REXX script via the input parameter. All default xml files are listed in "Chapter 9. Default XML files" on page 185. The parameters in these files are set to the default values of the SM Administration EUI.

The default xml file must be present in the path that is specified by the environment variable `DEFAULT_CLIENT_XML_PATH` or the user has to specify its path.

Example: `-xmlinput 'inputcreatelrm.xml'` specifies the default input xml file in the `DEFAULT_CLIENT_XML_PATH`. But `-xmlinput './inputcreatelrm.xml'` specifies the file in the current directory.

The user can modify the default path for the default xml files by setting the environment variable `DEFAULT_CLIENT_XML_PATH` to another existing path. Make sure that the path exists and that the default xml file that will be used is in this directory.

-input
This is an optional parameter for the CB390CFG API. It specifies the input file which contains the name value pairs that should override the parameters of the default xml file. To generate a xml file by using REXX variables there is a tool called XMLGEN. This tool is described in "Chapter 5. XMLGEN" on page 169.

Important: The input file will be deleted after the parameters are merged with the default xml file.

-output
The output file contains further information. In the description of each LRM action there is an example output file. The general output format for a LRM action looks like this:

```
administratorname.1 AdministratorName
coclasscreatefunction.1 CoClassCreateFunctionState
coclassname.1 CoClassName
codllname.1 CoDllName
conversationname.1 ConversationName
lrmdescription.1 LRMDescription
lrmname.1 LRMName
lrmsubsystemtype.1 LRMSubsystemTypeState
sysplexname.1 SysplexName
status 0|4
message.1 OK|ErrorMessage
count NumberOfListedLRM
```

Action “createlrm”

This action causes a new LRM to be created.

Syntax

```
►►──rc = CB390CFG──("── -action──'createlrm' ──────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬──────►
                                    └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")──────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for createlrm "inputcreatelrm.xml" is listed in section “inputcreatelrm.xml” on page 214. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputcreatelrm.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputcreatelrm.xml"`. If you want to use your own default xml file you **must** specify the

| complete directory of the file or you **must** set the
| `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file

The table below includes all of the attributes that are known for this LRM action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputcreatelrm.xml" on page 214.

Parameter name	Values	Required
conversationname	Name of the conversation "*" to list all conversations pertaining to the Administrator	x
lrmname	Name of the lrm	x
lrmdescription	Description of the lrm	
coclassname		
codllname		
coclasscreatefunction		
lrmsubsystemtype		

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "lrmname"
name.3 = "lrmdescription"
name.4 = "coclassname"
name.5 = "codllname"
name.6 = "coclasscreatefunction"
name.7 = "lrmsubsystemtype"
```

```
val. = 0
val.1 = "API Functiontest"
val.2 = "API_Functiontest_LRM"
val.3 = "API Functiontest LRM Description"
val.4 = ""
val.5 = ""
val.6 = ""
val.7 = "DB2"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #19 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'createlrm' -xmlinput 'inputcreatelrm.xml'
               -input 'tempin' -output 'FCT19'")
if (rc == 4) then do
  say "FCT Test #19 failed"
  exit
end
exit

error:
say "Error in FCT Test #19" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
coclasscreatefunction.1 DB2RRSAF390ResourceMgr_IResourceMgrAdminObject_Impl_Crea
coclassname.1 DB2RRSAF390ResourceMgr::IResourceMgrAdminObject
codllname.1 BBOIDRMI
conversationname.1 API Functiontest
lrmdescription.1 API Functiontest LRM Description
lrmname.1 API_Functiontest_LRM
lrmsubsystemtype.1 DB2
sysplexname.1 PLEX1
status 0
message.1 OK
count 1
```

Action “deletelrm”

This action causes the named LRM to be deleted. This is a logical deletion. The deletion does not occur until the conversation this change is associated with is commited.

Syntax

```
►►──rc = CB390CFG──("── -action──'deletlrm' ──────────────────────────►

►── -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬──────►
                                     └─ -input──'inputfilename' ─┘

►── -output──'outputfilename' ──")──────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for deletelrm "inputdeletelrm.xml" is listed in section "inputdeletelrm.xml" on page 214. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory. If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputdeletelrm.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputdeletelrm.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file

The table below includes all of the attributes that are known for this LRM action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputdeletelrm.xml" on page 214.

Parameter name	Values	Required
conversationname	Name of the conversation	x
lrmname	Name of the lrm	x

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "lrmname"

val. = 0
val.1 = "API Functiontest"
val.2 = "API_Functiontest_LRM"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #21 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'deletelrm' -xmlinput 'inputdeletelrm.xml'
               -input 'tempin' -output 'FCT21'")
if (rc == 4) then do
  say "FCT Test #21 failed"
  exit
end
exit

error:
say "Error in FCT Test #21" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
/* REXX function */
call syscalls ÆONÆ
signal on error
```

```
name. = 0
name.1 = "conversationname"
name.2 = "lrmname"
val. = 0
val.1 = "API Functiontest"
val.2 = "API_Functiontest_LRM"
rc = 4
i = 1
do while(name.i <> '0')

rc = XMLGEN("tempin" name.i val.i)
if (rc == 4) then do
say "FCT Test #21 failed while XMLGEN"
exit
end
i = i+1
end;
rc = CB390CFG("-action 'deletelrm' -xmlinput 'inputdeletelrm.xml'
-input 'tempin' -output 'FCT21'")
if (rc == 4) then do
say "FCT Test #21 failed"
exit
end
exit
error:
say "Error in FCT Test #21" rc "at line" sigl
say sourceline(sigl)
exit
```

Action “changelrm”

This action causes the attributes of the named LRM to be changed.

Syntax

```
rc = CB390CFG("-action 'changelrm'
              -xmlinput 'defaultxmlfilename'
              [-input 'inputfilename']
              -output 'outputfilename'")
```

```
►►──rc = CB390CFG──("── -action──'changelrm' ────────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬───────────────────────────┬───────►
                                     └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")──────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename

This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for changelrm "inputchangelrm.xml" is listed in section "inputchangelrm.xml" on page 215. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputchangelrm.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputchangelrm.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename

This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename

This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file

The table below includes all of the attributes that are known for this LRM action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputchangelrm.xml" on page 215.

Parameter name	**Values**	**Required**
conversationname	Name of the conversation	x
lrmname	Name of the lrm	x
lrmdescription	Description of the lrm	x
coclassname		
codllname		
coclasscreatefunction		
lrmsubsystemtype		

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "lrmname"
name.3 = "lrmdescription"
name.4 = "coclassname"
name.5 = "codllname"
name.6 = "coclasscreatefunction"
name.7 = "lrmsubsystemtype"

val. = 0
val.1 = "API Functiontest"
val.2 = "API_Functiontest_LRM"
val.3 = "API Functiontest LRM Description modified"
val.4 = ""
val.5 = ""
val.6 = ""
val.7 = "IMS_OTMA_PAA"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #20 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'changelrm' -xmlinput 'inputchangelrm.xml'
               -input 'tempin' -output 'FCT20'")
if (rc == 4) then do
  say "FCT Test #20 failed"
  exit
end
exit

error:
say "Error in FCT Test #20" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
coclasscreatefunction.1 DB2RRSAF390ResourceMgr_IResourceMgrAdminObject_Impl_Creat
coclassname.1 DB2RRSAF390ResourceMgr::IResourceMgrAdminObject
codllname.1 BBOIDRMI
```

```
conversationname.1 API Functiontest
lrmdescription.1 API Functiontest LRM Description modified
lrmname.1 API_Functiontest_LRM
lrmsubsystemtype.1 DB2
sysplexname.1 PLEX1
status 0
message.1 OK
count 1
```

Action “listlrm”

This action causes the named LRM to be listed. If the LRM name equals "*" then all LRMs will be listed.

Syntax

```
►►──rc = CB390CFG──("── -action──'listlrm' ──────────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬──────────────────────────────┬───►
                                     └─ -input──'inputfilename' ────┘

►─ -output──'outputfilename' ──")──────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for listlrm "inputlistlrm.xml" is listed in section “inputlistlrm.xml” on page 216. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory. If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputlistlrm.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputlistlrm.xml"`. If you want to use your own de-fault xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these

new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file
The table below includes all of the attributes that are known for this LRM action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputlistlrm.xml" on page 216.

Parameter name	Values	Required
conversationname	Name of the conversation	x
lrmname	Name of the lrm "*" to list all LRM in the conversation	x

Example
Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "lrmname"

val. = 0
val.1 = "API Functiontest"
val.2 = "API_Functiontest_LRM"

rc = 4
i = 1

do while(name.i <>'0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #22 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'listlrm' -xmlinput 'inputlistlrm.xml'
               -input 'tempin' -output 'FCT22'")
if (rc == 4) then do
  say "FCT Test #22 failed"
  exit
end
exit

error:
say "Error  in FCT Test #22" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
coclasscreatefunction.1 DB2RRSAF390ResourceMgr_IResourceMgrAdminObject_Impl_Crea
coclassname.1 DB2RRSAF390ResourceMgr::IResourceMgrAdminObject
codllname.1 BBOIDRMI
conversationname.1 API Functiontest
lrmdescription.1 API Functiontest LRM Description modified
lrmname.1 API_Functiontest_LRM
lrmsubsystemtype.1 DB2
sysplexname.1 PLEX1
status 0
message.1 OK
count 1
```

LRMI

These functions are for the modifications of a LRMI.

Syntax

```
►►—rc = CB390CFG—("— -action—'——┬create- ——┬lrmi—' ——————————————————►
                               ├delete- —┤
                               ├change- —┤
                               └list- ———┘

►— -xmlinput—'defaultxmlfilename' ——┬——————————————————————————┬——————►
                                    └ -input—'inputfilename' ——┘

►— -output—'outputfilename' —") ——————————————————————————————————————►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "0" if error has occurred while processing the action.

-action

createlrmi	Causes a new lrmi to be created.
deletelrmi	Causes the lrmi to be deleted.

changelrmi	Causes the lrmi to be changed.
listlrmi	Causes the lrmi to be listed.

-xmlinput
This is the default xml file. In this file all required parameters for the action which should be performed must be specified. This file is a xml file with a document type definition (DTD). The DTD only specifies the structure of the document. The user can specify default values for each parameter. These parameters can be overriden by the REXX script via the input parameter. All default xml files are listed in “Chapter 9. Default XML files” on page 185. The parameters in these files are set to the default values of the SM Administration EUI.

The default xml file must be present in the path that is specified by the environment variable `DEFAULT_CLIENT_XML_PATH` or the user has to specify its path.

Example: `-xmlinput 'inputcreatelrmi.xml'` specifies the default input xml file in the `DEFAULT_CLIENT_XML_PATH`. But `-xmlinput './inputcreatelrmi.xml'` specifies the file in the current directory.

The user can modify the default path for the default xml files by setting the environment variable `DEFAULT_CLIENT_XML_PATH` to another existing path. Make sure that the path exists and that the default xml file that will be used is in this directory.

-input
This is an optional parameter for the CB390CFG API. It specifies the input file which contains the name value pairs that should override the parameters of the default xml file. To generate a xml file by using REXX variables there is a tool called XMLGEN. This tool is described in “Chapter 5. XMLGEN” on page 169.

Important: The input file will be deleted after the parameters are merged with the default xml file.

-output
The output file contains further information. In the description of each LRMI action there is an example output file. The general output format for a LRMI action looks like this:

```
administratorname.1 AdministratorName
conversationname.1 ConversationName
lrmidescription.1 LRMIDescription
lrminame.1 LRMIName
lrmname.1 LRMName
sysplexname.1 SysplexName
```

```
systemname.1 SystemName
status 0|4
message.1 OK|ErrorMessage
count NumberOfListedLRMI
```

Action “createlrmi”

This action causes a new LRMI to be created.

Syntax

```
►►──rc = CB390CFG──("── -action──'createlrmi' ─────────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬───────────►
                                    └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")─────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for createlrmi "inputcreatelrmi.xml" is listed in section “inputcreatelrmi.xml” on page 217. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputcreatelrmi.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputcreatelrmi.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file
The table below includes all of the attributes that are known for this LRMI action. The required ones must be defined in the default xml file or can be defined by the XMLGEN (“Chapter 5. XMLGEN” on page 169) script. The default xml file is listed in section “inputcreatelrmi.xml” on page 217.

Parameter name	Values	Required
conversationname	Name of the conversation	x
lrmname	Name of the lrm	x
lrminame	Name of the lrmi	x
lrmidescription	Description of the lrmi	
systemname	Name of the system	x
connection	Connection data To specify the connection data in the default xml there must be set an element for each connection data type. If a connection data is specified the attributes **must** be set. To specify the connection data in a REXX script you must specify only one name value pair for all connection data. For the "name"value specify connection and into the "value"value insert the connection data like the following format: name='value' [name='value' ...]. The parameter name specifies the name of the connection data, value specifies its value.	

Example
Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "lrmname"
name.3 = "lrminame"
name.4 = "lrmidescription"
name.5 = "systemname"

val. = 0
val.1 = "API Functiontest"
val.2 = "API_Functiontest_LRM"
val.3 = "API_Functiontest_LRMI"
val.4 = "API Functiontest LRMI Description"
val.5 = "SY1"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #23 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'createlrmi' -xmlinput 'inputcreatelrmi.xml'
               -input 'tempin' -output 'FCT23'")
if (rc == 4) then do
  say "FCT Test #23 failed"
  exit
end
exit

error:
say "Error in FCT Test #23" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
conversationname.1 API Functiontest
lrmidescription.1 API Functiontest LRMI Description
lrminame.1 API_Functiontest_LRMI
lrmname.1 API_Functiontest_LRM
sysplexname.1 PLEX1
systemname.1 SY1
connection.1 ID1 = 'test1' ID2 = 'test2'
status 0
message.1 OK
count 1
```

Action “deletelrmi”

This action causes the named LRMI to be deleted. This is a logical deletion. The deletion does not occur until th conversation this change is associated with is commited.

Syntax

```
►►──rc = CB390CFG──("── -action──'deletelrmi' ──────────────────────────►
►─ -xmlinput─'defaultxmlfilename' ──┬──────────────────────────┬──►
                                    └─ -input─'inputfilename' ─┘

►─ -output─'outputfilename' ─")──────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for deletelrmi "inputdeletelrmi.xml" is listed in section "inputdeletelrmi.xml" on page 218. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputdeletelrmi.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputdeletelrmi.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file

The table below includes all of the attributes that are known for this LRMI action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputdeletelrmi.xml" on page 218.

Parameter name	Values	Required
conversationname	Name of the conversation	x
lrmname	Name of the lrm	x
lrminame	Name of the lrmi	x

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "lrmname"
name.3 = "lrminame"

val. = 0
val.1 = "API Functiontest"
val.2 = "API_Functiontest_LRM"
val.3 = "API_Functiontest_LRMI"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #25 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'deletelrmi' -xmlinput 'inputdeletelrmi.xml'
              -input 'tempin' -output 'FCT25'")
if (rc == 4) then do
  say "FCT Test #25 failed"
  exit
end
exit

error:
say "Error in FCT Test #25" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
conversationname.1 API Functiontest
lrmidescription.1 API Functiontest LRMI Description modified
lrminame.1 API_Functiontest_LRMI
```

```
lrmname.1 API_Functiontest_LRM
sysplexname.1 PLEX1
systemname.1 SY1
connection.1 ID2 = 'changedtest2' ID1 = 'changedtest1'
status 0
message.1 OK
count 1
```

Action “changelrmi”

This action causes the attributes for the named LRMI to be changed.

Syntax

```
►►──rc = CB390CFG──(″── -action──’changelrmi’ ──────────────────────────────►

►─ -xmlinput──’defaultxmlfilename’ ─┬──────────────────────────┬────────────►
                                    └─ -input──’inputfilename’ ─┘

►─ -output──’outputfilename’ ──″)──────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for changelrmi "inputchangelrmi.xml" is listed in section “inputchangelrmi.xml” on page 218. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory.

  If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputchangelrmi.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputchangelrmi.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
: This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these

new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file

The table below includes all of the attributes that are known for this LRMI action. The required ones must be defined in the default xml file or can be defined by the XMLGEN (“Chapter 5. XMLGEN” on page 169) script. The default xml file is listed in section “inputchangelrmi.xml” on page 218.

Parameter name	Values	Required
conversationname	Name of the conversation	x
lrmname	Name of the lrm	x
lrminame	Name of the lrmi	x
lrmidescription	Description of the lrmi	
systemname	Name of the system	x
connection	Connection data To specify the connection data in the default xml there must be set an element for each connection data type. If a connection data is specified the attributes must be set. To specify the connection data in a REXX script you must specify only one name value pair for all connection data. For the "name"value specify connection and into the "value"value insert the connection data like the following format: name='value' [name='value' ...]. The parameter name specifies the name of the connection data, value specifies its value.	

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "lrmname"
name.3 = "lrminame"
name.4 = "lrmidescription"
name.5 = "systemname"

val. = 0
val.1 = "API Functiontest"
val.2 = "API_Functiontest_LRM"
val.3 = "API_Functiontest_LRMI"
val.4 = "API Functiontest LRMI Description modified"
val.5 = "SY1"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #24 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'changelrmi' -xmlinput 'inputchangelrmi.xml'
               -input 'tempin' -output 'FCT24'")
if (rc == 4) then do
  say "FCT Test #24 failed"
  exit
end
exit

error:
say "Error in FCT Test #24" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
conversationname.1 API Functiontest
lrmidescription.1 API Functiontest LRMI Description modified
lrminame.1 API_Functiontest_LRMI
lrmname.1 API_Functiontest_LRM
sysplexname.1 PLEX1
systemname.1 SY1
connection.1 ID1 = 'changedtest1' ID2 = 'changedtest2'
status 0
message.1 OK
count 1
```

Action “listlrmi”

This action causes the named LRMI to be listed. If the LRMI name equals "*"then all LRMIs will be listed.

Syntax

```
►►──rc = CB390CFG──("── -action──'listlrmi' ─────────────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬──────────────────────────┬────►
                                     └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──") ─────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for listlrmi "inputlistlrmi.xml" is listed in section "inputlistlrmi.xml" on page 219. This file is present in the "/usr/lpp/WebSphere/samples/smapi" directory. If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputlistlrmi.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/WebSphere/samples/smapi/inputlistlrmi.xml"`. If you want to use your own de-fault xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file

The table below includes all of the attributes that are known for this LRMI action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputlistlrmi.xml" on page 219.

Parameter name	Values	Required
lrmname	Name of the lrm	x
lrminame	Name of the lrmi "*"to list all LRMI in associated with the LRM	

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "lrmname"
name.3 = "lrminame"

val. = 0
val.1 = "API Functiontest"
val.2 = "API_Functiontest_LRM"
val.3 = "API_Functiontest_LRMI"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #26 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'listlrmi' -xmlinput 'inputlistlrmi.xml'
               -input 'tempin' -output 'FCT26'")
if (rc == 4) then do
  say "FCT Test #26 failed"
  exit
end
exit

error:
say "Error in FCT Test #26" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
conversationname.1 API Functiontest
lrmidescription.1 API Functiontest LRMI Description modified
```

```
lrminame.1 API_Functiontest_LRMI
lrmname.1 API_Functiontest_LRM
sysplexname.1 PLEX1
systemname.1 SY1
connection.1 ID2 = 'changedtest2' ID1 = 'changedtest1'
status 0
message.1 OK
count 1
```

Container/LRM associations

These functions are for the modifications of the associations between LRMs and containers.

Syntax

```
►►──rc = CB390CFG──("── -action──'──┬─associatelrmwithcontainer─────┬─'────►
                                    ├─disassociatelrmfromcontainer──┤
                                    └─listlrmassociatedwithcontainer┘

►─ -xmlinput──'defaultxmlfilename' ──┬───────────────────────────┬─────────►
                                     └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "0" if error has occurred while processing the action.

-action

associatelrmwithcontainer

disassociatelrmfromcontainer

listlrmassociatedwithcontainer

-xmlinput
This is the default xml file. In this file all required parameters for the action which should be performed must be specified. This file is a xml file with a document type definition (DTD). The DTD only specifies the structure of the document. The user can specify default values for each parameter. These parameters can be overriden by the REXX script via the input parameter. All default xml files are listed in “Chapter 9. Default XML files” on page 185. The parameters in these files are set to the default values of the SM Administration EUI.

The default xml file must be present in the path that is specified by the environment variable `DEFAULT_CLIENT_XML_PATH` or the user has to specify its path.

Example: `-xmlinput 'inputassociatelrmwithcontainer.xml'` specifies the default input xml file in the `DEFAULT_CLIENT_XML_PATH`.
But `-xmlinput './inputassociatelrmwithcontainer.xml'` specifies the file in the current directory.

The user can modify the default path for the default xml files by setting the environment variable `DEFAULT_CLIENT_XML_PATH` to another existing path. Make sure that the path exists and that the default xml file that will be used is in this directory.

-input
This is an optional parameter for the CB390CFG API. It specifies the input file which contains the name value pairs that should override the parameters of the default xml file. To generate a xml file by using REXX variables there is a tool called XMLGEN. This tool is described in "Chapter 5. XMLGEN" on page 169.

Important: The input file will be deleted after the parameters are merged with the default xml file.

-output
The output file contains further information. In the description of each association action there is an example output file. The general output format for a association action looks like this:

```
administratorname.1 AdministratorName
containername.1 ContainerName
conversationname.1 ConversationName
lrmname.1 LRMName
servername.1 ServerName
sysplexname.1 SysplexName
status 0|4
message.1 OK|ErrorMessage
count NumberOfListedAssociations
```

Action "associatelrmwithcontainer"

This action causes an association of the named LRM with the named container.

Syntax

```
►►──rc = CB390CFG──("── -action──'associatelrmwithcontainer' ─────────────────►
►─ -xmlinput─'defaultxmlfilename' ─┬──────────────────────────┬──────►
                                    └─ -input─'inputfilename' ─┘

►─ -output─'outputfilename' ─")──────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for associatelrmwithcontainer "inputassociatelrmwithcontainer.xml" is listed in section "inputassociatelrmwithcontainer.xml" on page 220. This file is present in the "/usr/lpp/CB390/samples/smapi"directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputassociatelrmwithcontainer.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/CB390/samples/smapi/inputassociatelrmwithcontainer.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file

The table below includes all of the attributes that are known for this association action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputassociatelrmwithcontainer.xml" on page 220.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	x
lrmname	Name of the lrm	x
containername	Name of the container	x

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "lrmname"
name.4 = "containername"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"
val.3 = "API_Functiontest_LRM"
val.4 = "API_Functiontest_Container"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #30 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'associatelrmwithcontainer'
               -xmlinput 'inputassociatelrmwithcontainer.xml'
               -input 'tempin' -output 'FCT30'")
if (rc == 4) then do
  say "FCT Test #30 failed"
  exit
end
exit

error:
say "Error in FCT Test #30" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
containername.1 API_Functiontest_Container
conversationname.1 API Functiontest
lrmname.1 API_Functiontest_LRM
servername.1 APIFCT
sysplexname.1 PLEX1
status 0
message.1 OK
count 1
```

Action “disassociatelrmfromcontainer”

This action causes a disassociation of the named LRM from the named container.

Syntax

```
►►──rc = CB390CFG──("── -action──'disassociatelrmfromcontainer' ──────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬───────────────►
                                    └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for disassociatelrmfromcontainer "inputdisassociatelrmfromcontainer. xml"is listed in section “inputdisassociatelrmfromcontainer.xml” on page 221. This file is present in the "/usr/lpp/CB390/samples/smapi"directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputdisassociatelrmfromcontainer.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/CB390/samples/smapi/inputdisassociatelrmfromcontainer.xml"` If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file

The table below includes all of the attributes that are known for this association action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputdisassociatelrmfromcontainer.xml" on page 221.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	x
lrmname	Name of the lrm	x
containername	Name of the container	x

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "lrmname"
name.4 = "containername"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"
val.3 = "API_Functiontest_LRM"
val.4 = "API_Functiontest_Container"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
```

```
    say "FCT Test #31 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'disassociatelrmfromcontainer'
               -xmlinput 'inputdisassociatelrmfromcontainer.xml'
               -input 'tempin' -output 'FCT31'")
if (rc == 4) then do
  say "FCT Test #31 failed"
  exit
end
exit

error:
say "Error in FCT Test #31" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
containername.1 API_Functiontest_Container
conversationname.1 API Functiontest
lrmname.1 API_Functiontest_LRM
servername.1 APIFCT
sysplexname.1 PLEX1
status 0
message.1 OK
count 1
```

Action “listlrmassociatedwithcontainer”

This action causes the named LRM associated with the named container to be listet. If the LRM name equals "*"then all LRM associated with the container will be listed.

Syntax

```
►►──rc = CB390CFG──("── -action──'listlrmassociatedwithcontainer' ──────────►

►─ -xmlinput──'defaultxmlfilename' ──┬────────────────────────┬──────────►
                                     └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename

This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for listlrmassociatedwithcontainer "inputlistlrmassociatedwithcontainer.xml"is listed in section "inputlistlrmassociatedwithcontainer.xml" on page 221. This file is present in the "/usr/lpp/CB390/samples/smapi"directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputlistlrmassociatedwithcontainer.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/CB390/samples/smapi/inputlistlrmassociatedwithcontainer.xml"` If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename

This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename

This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file

The table below includes all of the attributes that are known for this association action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputlistlrmassociatedwithcontainer.xml" on page 221.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	x
lrmname	Name of the lrm "*" to list all LRM associated with the spec-ified container	x
containername	Name of the container	x

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "lrmname"
name.4 = "containername"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"
val.3 = "API_Functiontest_LRM"
val.4 = "API_Functiontest_Container"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #32 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'listlrmassociatedwithcontainer'
               -xmlinput 'inputlistlrmassociatedwithcontainer.xml'
               -input 'tempin' -output 'FCT32'")
if (rc == 4) then do
  say "FCT Test #32 failed"
  exit
end
exit

error:
say "Error in FCT Test #32" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
containername.1 API_Functiontest_Container
conversationname.1 API Functiontest
lrmname.1 API_Functiontest_LRM
servername.1 APIFCT
sysplexname.1 PLEX1
status 0
message.1 OK
count 1
```

Application family

Syntax

```
►►──rc = CB390CFG──("── -action──'──┬─import- ─┬──Applicationfamily──'──────────►
                                    ├─remove- ─┤
                                    └─list- ───┘

►── -xmlinput──'defaultxmlfilename' ──┬─────────────────────────────┬────────────►
                                      └─ -input──'inputfilename' ──┘

►── -output──'outputfilename' ──") ────────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "0" if error has occurred while processing the action.

-action

importApplicationfamily	Causes the application families to be configured to the server.
removeApplicationfamily	Causes causes the application families to be deleted.
listApplicationfamily	Causes the application families configured to a server to be listed.

-xmlinput
This is the default xml file. In this file all required parameters for the action which should be performed must be specified. This file is a xml file with a document type definition (DTD). The DTD only specifies the structure of the document. The user can specify default values for each parameter. These parameters can be overriden by the REXX script via the input parameter. All default xml files are listed in “Chapter 9. Default XML files” on page 185. The parameters in these files are set to the default values of the SM Administration EUI.

The default xml file must be present in the path that is specified by the environment variable `DEFAULT_CLIENT_XML_PATH` or the user has to specify its path.

Example: `-xmlinput 'inputimportApplicationfamily.xml'` specifies the default input xml file in the

`DEFAULT_CLIENT_XML_PATH`.
But -xmlinput './inputimportApplicationfamily.xml' specifies the file in the current directory.

The user can modify the default path for the default xml files by setting the environment variable `DEFAULT_CLIENT_XML_PATH` to another existing path. Make sure that the path exists and that the default xml file that will be used is in this directory.

-input
This is an optional parameter for the CB390CFG API. It specifies the input file which contains the name value pairs that should override the parameters of the default xml file. To generate a xml file by using REXX variables there is a tool called XMLGEN. This tool is described in "Chapter 5. XMLGEN" on page 169.

Important: The input file will be deleted after the parameters are merged with the default xml file.

-output
The output file contains further information. In the description of each applicaiton family action there is an example output file. The general output format for a application family action looks like this:

```
administratorname.1 AdministratorName
applicationfamilydescription.1 ApplicationFamilyDescription
applicationfamilyname.1 ApplicationFamilyName
conversationname.1 ConversationName
servername.1 ServerName
sysplexname.1 SysplexName
status 0|4
message.1 OK|ErrorMessage
count NumberOfListedApplicationFamilies
```

Action "importApplicationfamily"

This action causes the named application family to be imported.

Syntax

```
►►──rc = CB390CFG──("── -action──'importApplicationfamily' ──────────────────►

►── -xmlinput──'defaultxmlfilename' ─┬──────────────────────────┬────────────►
                                     └─ -input──'inputfilename' ─┘

►── -output──'outputfilename' ──") ───────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for importApplicationfamily "inputimportApplicationfamily. xml"is listed in section "inputimportapplicationfamily.xml" on page 222. This file is present in the "/usr/lpp/CB390/samples/smapi"directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputimportApplicationfamily.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/CB390/samples/smapi/inputimportApplicationfamily.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file
The table below includes all of the attributes that are known for this application family action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputimportapplicationfamily.xml" on page 222.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	x
applicationfamilyname	Name of the application family	x
ddlfilename	Name of the dll file dataset	x
outputfilename	Name of the output dataset	x

Example

Here is an example script:

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "applicationfamilyname"
name.4 = "ddlfilename"
name.5 = "outputfilename"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"
val.3 = "API_Functiontest_Application"
val.4 = "BOSSMN3.DDL(DEMO)"
val.5 = "BOSSMN3.GUIOUT.ERRLOG"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #27 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'importApplicationfamily'
               -xmlinput 'inputimportApplicationfamily.xml'
               -input 'tempin' -output 'FCT27'")
if (rc == 4) then do
  say "FCT Test #27 failed"
  exit
end
exit

error:
say "Error in FCT Test #27" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
applicationfamilydescription.1
applicationfamilyname.1 Scripting
conversationname.1 API Functiontest
servername.1 APIFCT
```

```
sysplexname.1 PLEX1
status 0
message.1 OK
count 1
```

Action “removeApplicationfamily”

This action causes the named application family to be deleted. This is a logical deletion. The deletion does not occur until the conversation this change is associated with is commited.

Syntax

```
►►──rc = CB390CFG──("── -action──'removeApplicationfamily' ─────────────────►

►─ -xmlinput──'defaultxmlfilename' ──┬──────────────────────────┬──────────►
                                      └─ -input──'inputfilename' ─┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for removeApplicationfamily "inputremoveApplicationfamily. xml"is listed in section “inputremoveapplicationfamily.xml” on page 223. This file is present in the "/usr/lpp/CB390/samples/smapi"directory.

  If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputremoveApplicationfamily.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/CB390/samples/smapi/inputremoveApplicationfamily.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
: This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN (“Chapter 5. XMLGEN” on page 169) you can set the values of the default xml file to these

new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file
The table below includes all of the attributes that are known for this application family action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputremoveapplicationfamily.xml" on page 223.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	
applicationfamilyname	Name of the application family	

Example
Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "applicationfamilyname"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"
val.3 = "API_Functiontest_Application"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #29 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'removeApplicationfamily'
              -xmlinput 'inputremoveApplicationfamily.xml'
```

```
               -input 'tempin' -output 'FCT29'")
if (rc == 4) then do
  say "FCT Test #29 failed"
  exit
end
exit

error:
say "Error in FCT Test #29" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
applicationfamilydescription.1
applicationfamilyname.1 Scripting
conversationname.1 API Functiontest
servername.1 APIFCT
sysplexname.1 PLEX1
status 0
message.1 OK
count 1
```

Action “listApplicationfamily”

This action causes the named application family to be listed. If the family application name equals "*"then all application families will be listed.

Syntax

```
►►──rc = CB390CFG──("── -action──'listApplicationfamily' ──────────────────────►

►─ -xmlinput──'defaultxmlfilename' ─┬─────────────────────────────┬─────────────►
                                    └─ -input──'inputfilename' ──┘

►─ -output──'outputfilename' ──")──────────────────────────────────────────────►◄
```

Syntax details

rc The return code (rc) is "0" if everything ended correctly. If rc is "0" if error has occurred while processing the action.

defaultxmlfilename
: This is the default xml file. The file has to contain a document type definition (DTD) and all of the required parameters. Only the optional attributes can be left out. The default xml file for listApplicationfamily "inputlistApplicationfamily.xml" is listed in section “inputlistapplicationfamily.xml” on page 224. This file is present in the "/usr/lpp/CB390/samples/smapi"directory.

If the environment variable `DEFAULT_CLIENT_XML_PATH` locates to this directory you only need to type the filename `"inputlistApplicationfamily.xml"`. Otherwise specify the complete location to the default xml file by setting this parameter to `"/usr/lpp/CB390/samples/smapi/inputlistApplicationfamily.xml"`. If you want to use your own default xml file you **must** specify the complete directory of the file or you **must** set the `DEFAULT_CLIENT_XML_PATH` to this directory.

inputfilename
This parameter is optional. It specifies a file that contains only name value pairs. Using XMLGEN ("Chapter 5. XMLGEN" on page 169) you can set the values of the default xml file to these new specified values. An example below show how this works. If it is not present, the default xmlinput file **must** contain all of the required parameters.

outputfilename
This parameter specifies the name of the output file. It will be written into the "/tmp"directory.

Values of default xml file

The table below includes all of the attributes that are known for this application family action. The required ones must be defined in the default xml file or can be defined by the XMLGEN ("Chapter 5. XMLGEN" on page 169) script. The default xml file is listed in section "inputlistapplicationfamily.xml" on page 224.

Parameter name	Values	Required
conversationname	Name of the conversation	x
servername	Name of the server	x
applicationfamilyname	Name of the application family "*" to list all application families in the specified server	x

Example

Here is an example script :

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "applicationfamilyname"

val. = 0
val.1 = "API Functiontest"
val.2 = "APIFCT"
val.3 = "API_Functiontest_Application"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "FCT Test #28 failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'listApplicationfamily'
               -xmlinput 'inputlistApplicationfamily.xml'
               -input 'tempin' -output 'FCT28'")
if (rc == 4) then do
  say "FCT Test #28 failed"
  exit
end
exit

error:
say "Error in FCT Test #28" rc "at line" sigl
say sourceline(sigl)
exit
```

The output file may look like this:

```
administratorname.1 CBADMIN
applicationfamilydescription.1
applicationfamilyname.1 Scripting
conversationname.1 API Functiontest
servername.1 APIFCT
sysplexname.1 PLEX1
status 0
message.1 OK
count 1
```

Chapter 5. XMLGEN

This REXX script is for generating the input file for the CB390CFG script. If you ever want to change some of the default values of the default xml files then you have to use a function like this to change the attributes value. This function opens a file and writes the value into it. For merging this values with the one out of the default xml files just call the administration function with the -input

```
'tempinputfilename'
```

parameter.

Syntax

```
►►──rc = XMLGEN──(──"tempinputfilename"──attributename──attributevalue──)──────►◄
```

Syntax Details

rc Return code from performed operation. This signals if the operation ended successfully (rc = 0) or if an error occurred (rc = 4).

tempinputfilename
This is the name of the input file. This input file is only temporarily available. It can be used for input for the CB390CFG tool ("Chapter 4. CB390CFG" on page 17). The file will be written to the /tmp directory.

attributename
This is the name of the attribute that should be changed.

attributevalue
This is the value of the attribute that should be set.

XMLGEN Script Code

For a better understanding of what this function does the code of the XMLGEN script is listed.

```
| /* REXX ---------------------------------------------------------------- */
| /* ===================================================================== */
| /*                                                                       */
| /* COPYRIGHT =                                                           */
| /* Licensed Material - Property of IBM                                   */
| /*                                                                       */
| /* 5655-A98 (C) Copyright IBM Corp. 2000                                 */
| /* All Rights Reserved.                                                  */
| /* U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or */
```

```
/* Disclosure restricted by GSA-ADP schedule contract with IBM Corp.*/
/* Status = H28K510                                                 */
/*                                                                  */
/* FILENAME: XMLGEN                                                 */
/*                                                                  */
/*                                                                  */
/* FUNCTION:                                                        */
/* REXX script for generating the tempin input file for             */
/* SM Scripting API                                                 */
/*                                                                  */
/* ================================================================ */
/* This script opend the specified file "filename" and writed the   */
/* "name" and the "val" into it. If an error occured the script     */
/* returns "4". Otherwise "0"                                       */

parse arg filename name val

path="/tmp/"||filename

ADDRESS SYSCALL

"open" path,
            O_rdwr+O_creat+O_append,
                777

if (retval == (-1)) then do
  say 'file not opened, error codes' errno errnojr ERRORTEXT(errno)
  return 4
end

fd = retval
name = name || " "
val = val || esc_n
'write' fd 'name' length(name)
'write' fd 'val' length(val)

if (retval == (-1)) then do
  say 'record not written, error codes' errno errnojr 'close' fd
  return 4
end

'close' fd

return 0

exit
```

Example

This example script writes the attributename conversationname with the value Document Demo and another attribute named conversationdescription with the value Document Demo Description into the file \ tmp \ tempin and calls following the createconversation script of the administration tool.

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "conversationdescription"

val. = 0
val.1 = "Document Demo"
val.2 = "Document Demo Description"

rc=0
i=1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "Error in function: createconversation while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'createconversation' -xmlinput 'inputcreateconversation.x
               -input 'tempin' -output 'tempout'")
if (rc == 4) then
  say "Error in function: createconversation"
exit

error:
say "Error" rc "at line" sigl
say sourceline(sigl)
exit
```

Chapter 6. XMLPARSE

This REXX script is for printing out the result. If the user wants to see the result parameter from the function call, he or she can look into the output file or print the result out by using such a REXX script. This function opens the specified file and prints it out line by line. The format of the file must be

```
attributename value
```

for each line. All output files of the cb390cfg and cb390cmd functions are already in this format. An example below shows how the XMLPARSE script works.

Syntax

```
►►──rc = XMLPARSE──(──"filename"──"target"──)────────────────────────────►◄
```

Syntax Details

rc The return code (rc) is "0" if everything ended correctly. If the return code (rc) is "4" an error has occurred while processing the action.

filename
This is the filename where the parser should find the target. If this file is not valid or if the target is not found the function returns "4".

target
This is the description of the target. Therefore three values are allowed:

V	Displays the values only.
N	Displays the attribute names only.
ALL	Displays the attribute names with their values.

XMLPARSE Script Code
For a better understanding of what this function does the code of the REXX script XMLPARSE is listed.

```
/* REXX -------------------------------------------------------------- */
/* ================================================================== */
/*                                                                    */
/* COPYRIGHT =                                                        */
/* Licensed Material - Property of IBM                                */
/*                                                                    */
/* 5655-A98 (C) Copyright IBM Corp. 2000                              */
/* All Rights Reserved.                                              */
/* U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or */
/* Disclosure restricted by GSA-ADP schedule contract with IBM Corp.*/
/* Status = H28K510                                                  */
/*                                                                   */
/* FILENAME: XMLPARSE                                                */
/*                                                                   */
/*                                                                   */
/* FUNCTION:                                                         */
/* REXX script to list values on the screen                          */
/* ================================================================= */
/* This script opens the specified file "filename" for reading      */
/* and displays for all lines in the file the specified information.*/
/* The type of the information can be set by the "nameorvalue"      */
/* parameter. "N" displays the names only. "V" displays the values  */
/* only. "ALL" displays both (name and value).                       */

parse arg filename nameorvalue

filename = "/tmp/"||filename

ADDRESS SYSCALL

readfile filename info.
ADDRESS

do i=1 to info.0
  parse var info.i var_name var_value
  if nameorvalue = "V" then
    say var_value
  if nameorvalue = "N" then
    say var_name
  if nameorvalue = "ALL" then
    say var_name " = " var_value
end

return 0
exit
```

Example

This example script writes the parameter into the input file ("Chapter 5. XMLGEN" on page 169) calls the function createconversation with the specified files (-input tempin and -output tempout) and parses the tempout file to print out all attributes.

```
/* REXX function */
call syscalls 'ON'
signal on error

name. = 0
name.1 = "conversationname"
name.2 = "conversationdescription"

val. = 0
val.1 = "Document Demo"
val.2 = "Document Demo Description"

rc = 4
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "Error in function: createconversation while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'createconversation' -xmlinput 'inputcreateconversation.x
               -input 'tempin' -output 'tempout'")
if (rc == 4) then do
  say "Error in function: createconversation"
  exit
end

error:
say "Error" rc "at line" sigl
say sourceline(sigl)
exit
```

Chapter 7. XMLFIND

This REXX script is for finding a special attribute in a file. The format of the file must be attributename value for each line of the file. This funciton is similar to the XMLEXTRACT script ("Chapter 8. XMLEXTRACT" on page 181) but the XMLFIND script can only return the value of a known attribute.

Syntax

```
►►──rc = XMLFIND──(──"filename"──"attributename"──)──────────────────────►◄
```

Syntax Details

data
: This is the value of the specified attribute.

filename
: This is the filename where the script looks into to find the specified attribute.

attributename
: This is the attributes name the script should find in the specified file.

XMLFIND Script Code

For a better understanding of what this function does the code of the REXX script XMLFIND is listed.

```
/* REXX ------------------------------------------------------------- */
/* ================================================================== */
/*                                                                    */
/* COPYRIGHT =                                                        */
/* Licensed Material - Property of IBM                                */
/*                                                                    */
/* 5655-A98 (C) Copyright IBM Corp. 2000                              */
/* All Rights Reserved.                                               */
/* U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or */
/* Disclosure restricted by GSA-ADP schedule contract with IBM Corp.*/
/* Status = H28K510                                                   */
/*                                                                    */
/* FILENAME: XMLFIND                                                  */
/*                                                                    */
/*                                                                    */
/* FUNCTION:                                                          */
/* REXX script for getting the value of the specified attribute     */
/*                                                                    */
/* ================================================================== */
/* This script opens the specified file "filename" for reading      */
/* and searches for the specified "namevalue" in the file.          */
/* If the attribute is found the value is returned. Otherwise       */
/* "4" is returned                                                  */

parse arg filename namevalue

ADDRESS SYSCALL

readfile "/tmp/"||filename info.
ADDRESS
do i=0 to info.0
  parse var info.i var_name var_value
  if var_name = namevalue then
    return var_value
end

return 4
EXIT
```

Example

This example script writes the parameter into the input file ("Chapter 5. XMLGEN" on page 169) calls the function createconversation with the specified files (-input tempin and -output tempout) and parses the tempout file for the attribute conversationname and uses this attributes and its value to generate a new input file. This new input file is used to perform the function commitconversation.

```
/* REXX ------------------------------------------------------------ */
/* ================================================================= */
/*                                                                   */
/* COPYRIGHT =                                                       */
/* Licensed Material - Property of IBM                               */
/*                                                                   */
/* 5655-A98 (C) Copyright IBM Corp. 2000                             */
/* All Rights Reserved.                                              */
/* U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or */
/* Disclosure restricted by GSA-ADP schedule contract with IBM Corp.*/
/* Status = H28K510                                                  */
/*                                                                   */
/* FILENAME: XMLFIND                                                 */
/*                                                                   */
/*                                                                   */
/* FUNCTION:                                                         */
/* REXX script for getting the value of the specified attribute     */
/*                                                                   */
/* ================================================================= */
/* This script opens the specified file "filename" for reading      */
/* and searches for the specified "namevalue" in the file.          */
/* If the attribute is found the value is returned. Otherwise       */
/* "4" is returned                                                  */

parse arg filename namevalue
```

```
ADDRESS SYSCALL

readfile "/tmp/"||filename info.
ADDRESS
do i=0 to info.0
  parse var info.i var_name var_value
  if var_name = namevalue then
    return var_value
end

return 4
EXIT
```

Chapter 8. XMLEXTRACT

This REXX script extracts from special line in the specified file the attribute name or the value. This script can be used to read an output file of an other REXX script line by line. This function is similar to the XMLFIND script ("Chapter 7. XMLFIND" on page 177) but the XMLEXTRACT script can read for REXX unknown attributes or values out of the input file.

Syntax

```
►►──rc = XMLEXTRACT──(──"inputfilename"──"line target"──)──────────────────►◄
```

Syntax Details

data
: Specifies the returned data from the script this can be an attribute name or a value.

inputfilename
: This is the name of the input file. The format of the file must be name value for each line.

line
: This is the line where the script takes its data.

target
: This specifies what the script should be extract, the name or the value. Expected values are

N	For the name of the attribute which is found in the specified line will be extracted.
V	For the value of the attribute which is found in the specified line will be extracted.

XMLGEN Script Code

For a better understanding of what this function does the code of the XMLEXTRACT script is listed.

```
/* REXX ----------------------------------------------------------- */
/* ================================================================ */
/*                                                                  */
/* COPYRIGHT =                                                      */
/* Licensed Material - Property of IBM                              */
/*                                                                  */
/* 5655-A98 (C) Copyright IBM Corp. 2000                            */
/* All Rights Reserved.                                             */
/* U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or */
/* Disclosure restricted by GSA-ADP schedule contract with IBM Corp.*/
/* Status = H28K510                                                 */
/*                                                                  */
/* FILENAME: XMLEXTRACT                                             */
/*                                                                  */
/*                                                                  */
/* FUNCTION:                                                        */
/* REXX script for getting the value or the specified attribute name*/
/* of a special line in the spacified file                          */
/* ================================================================ */
/* This script opens the specified file "filename" for reading      */
/* and returns the name or the value of the specified line in the   */
/* file "filename". What the script should return can be specified  */
/* with the "nameorvalue" flag. "V" is for value, "N" is for name.  */

parse arg filename line nameorvalue

ADDRESS SYSCALL

readfile "/tmp/"||filename info.

ADDRESS

parse var info.line var_name var_value

if var_name = 'status' then
   return 0
if nameorvalue = "V" then
   return var_value
if nameorvalue = "N" then
   parse var var_name var_name '.' var_nr
   return var_name

exit
```

Example

This example script will change the settings of a server. First the server which should be modified will be listed. The output will be written into the tempout file. XMLEXTRACT gets the attributes name and its value will be changed if this is desired. After that the tempout file will be updated by XMLGEN. If all changes have taken place the script calls the CB390CFG script.

```
/* REXX function */
call syscalls 'ON'
signal on error

sval. = 0
sname. = 0

name. = 0
name.1 = "conversationname"
name.2 = "servername"

val. = 0
val.1 = "Document Demo"
val.2 = "DEMOSRV"

rc = 4
i = 1
l = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "Error in Function: listserver while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'listserver' -xmlinput 'inputlistserver.xml'
               -input 'tempin' -output 'tempout'")
if (rc == 4) then do
  say "Error in Function: listserver"
  exit
end

do forever
  n = XMLEXTRACT("tempout" l "N")
  if n <> '0' then do
    sname.l = n
    if n = "serverdescription" then do
      sval.l = "New Description"
    end
    else if n = "garbagecollectioninterval" then do
      sval.l = "55555"
    end
    else do
      v = XMLEXTRACT("tempout" l "V")
      sval.l = v
    end
    rc = XMLGEN("tempin" sname.l sval.l)
    if (rc == 4) then do
      say "Error in Function: listserver while XMLGEN"
      exit
    end
  end
  else
    leave
  l=l+1
end

rc = CB390CFG("-action 'changeserver' -xmlinput 'inputchangeserver.xml'
               -input 'tempin' -output 'tempout'")
if (rc == 4) then do
  say "Error in Function: changeserver"
  exit
end
say "Server changed"
```

```
exit

error:
say "Error" rc "at line" sigl
say sourceline(sigl)
exit
```

Chapter 9. Default XML files

These files can be modified by the user. The values which are listed here are the default values of the SM-EUI. These can only be defined for the create methods. All other methods have their own parameter which must be specified by the user.

inputcreateconversation.xml

```
<?xml version='1.0'?>
<!--========================================================================-->
<!--   File name:  inputcreateconversation.xml                              -->
<!--                                                                        -->
<!--   Descriptive name:  ...                                               -->
<!--                                                                        -->
<!--   Proprietary statement:                                               -->
<!--                                                                        -->
<!-- Licensed Material - Property of IBM                                    -->
<!--                                                                        -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                  -->
<!-- All Rights Reserved.                                                   -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or       -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.      -->
<!-- Status = H28K510                                                       -->
<!--                                                                        -->
<!--  Change history:                                                       -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                          -->
<!--                                                                        -->
<!--========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputcreateconversation [
<!ELEMENT inputcreateconversation EMPTY>
<!ATTLIST inputcreateconversation
  conversationname CDATA #REQUIRED
  conversationdescription CDATA #IMPLIED
>
]>

<!--begin of default values-->
<inputcreateconversation
  conversationname = ''
  conversationdescription = ''
/>
```

inputdeleteconversation.xml

```
<?xml version='1.0'?>
<!--=======================================================================-->
<!--   File name:  inputdeleteconversation.xml                           -->
<!--                                                                     -->
<!--   Descriptive name:  ...                                            -->
<!--                                                                     -->
<!--   Proprietary statement:                                            -->
<!--                                                                     -->
<!-- Licensed Material - Property of IBM                                 -->
<!--                                                                     -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                               -->
<!-- All Rights Reserved.                                                -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or    -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.   -->
<!-- Status = H28K510                                                    -->
<!--                                                                     -->
<!--  Change history:                                                    -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                       -->
<!--                                                                     -->
<!--=======================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputdeleteconversation [
<!ELEMENT inputdeleteconversation EMPTY>
<!ATTLIST inputdeleteconversation
  conversationname CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputdeleteconversation
  conversationname = ''
/>
```

inputcommitconversation.xml

```
<?xml version='1.0'?>
<!--=======================================================================-->
<!--   File name:  inputcommitconversation.xml                           -->
<!--                                                                     -->
<!--   Descriptive name:  ...                                            -->
<!--                                                                     -->
<!--   Proprietary statement:                                            -->
<!--                                                                     -->
<!-- Licensed Material - Property of IBM                                 -->
<!--                                                                     -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                               -->
<!-- All Rights Reserved.                                                -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or    -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.   -->
<!-- Status = H28K510                                                    -->
<!--                                                                     -->
<!--  Change history:                                                    -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                       -->
```

```
<!--                                                                      -->
<!--======================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputcommitconversation [
<!ELEMENT inputcommitconversation EMPTY>
<!ATTLIST inputcommitconversation
  conversationname CDATA #REQUIRED
>
]>

<!--begin of dafeult values-->
<inputcommitconversation
  conversationname = ''
/>
```

inputlistconversation.xml

```
<?xml version='1.0'?>
<!--======================================================================-->
<!--   File name:  inputlistconversation.xml                              -->
<!--                                                                      -->
<!--   Descriptive name:  ...                                             -->
<!--                                                                      -->
<!--   Proprietary statement:                                             -->
<!--                                                                      -->
<!-- Licensed Material - Property of IBM                                  -->
<!--                                                                      -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                -->
<!-- All Rights Reserved.                                                 -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or     -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.    -->
<!-- Status = H28K510                                                     -->
<!--                                                                      -->
<!--  Change history:                                                     -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                        -->
<!--                                                                      -->
<!--======================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputlistconversation [
<!ELEMENT inputlistconversation EMPTY>
<!ATTLIST inputlistconversation
  conversationname CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputlistconversation
  conversationname = ''
/>
```

inputchangesysplex.xml

```
<?xml version='1.0'?>
<!--=======================================================================-->
<!--   File name:  inputchangesysplex.xml                                   -->
<!--                                                                        -->
<!--   Descriptive name:  ...                                               -->
<!--                                                                        -->
<!--   Proprietary statement:                                               -->
<!--                                                                        -->
<!-- Licensed Material - Property of IBM                                    -->
<!--                                                                        -->
<!-- 5655-F31 (C) Copyright IBM Corp. 2000, 2001                            -->
<!-- All Rights Reserved.                                                   -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or       -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.      -->
<!-- Status = H28W400                                                       -->
<!--                                                                        -->
<!--  Change history:                                                       -->
<!--$L0=OW44455, CB4.0_beta, 20001205, PDBL: Created.                       -->
<!--                                                                        -->
<!--=======================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputchangesysplex [
<!ELEMENT environment EMPTY>
<!ATTLIST environment
 value CDATA #REQUIRED
 name CDATA #REQUIRED
>
<!ELEMENT inputchangesysplex (environment*)>
<!ATTLIST inputchangesysplex
  conversationname CDATA #REQUIRED
  sysplexname CDATA #REQUIRED
  sysplexdescription CDATA #IMPLIED
  logstreamname CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputchangesysplex
  conversationname = ''
  sysplexname = ''
  sysplexdescription = ''
  logstreamname = ''
>
</inputchangesysplex>
```

inputlistsysplex.xml

```
<?xml version='1.0'?>
<!--=======================================================================-->
<!--   File name:  inputlistsysplex.xml                             -->
<!--                                                                        -->
<!--   Descriptive name:  ...                                               -->
```

```
<!--                                                                     -->
<!--                                                                     -->
<!--   Proprietary statement:                                            -->
<!--                                                                     -->
<!--   Licensed Material - Property of IBM                               -->
<!--   5655-F31 (C) Copyright IBM Corp. 2000, 2001                       -->
<!--                                                                     -->
<!--   All Rights Reserved.                                              -->
<!--                                                                     -->
<!--   U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or  -->
<!--   Disclosure restricted by GSA-ADP schedule contract with IBM Corp. -->
<!--                                                                     -->
<!--   Status = H28W400                                                  -->
<!--                                                                     -->
<!--                                                                     -->
<!--  Change history:                                                    -->
<!--$L0=OW44455, H28K510, 20001205, PDBL: Created.                       -->
<!--                                                                     -->
<!--=====================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputlistsysplex [
<!ELEMENT inputlistsysplex EMPTY>
<!ATTLIST inputlistsysplex
  conversationname CDATA #REQUIRED
  sysplexname CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputlistsysplex
  conversationname = ''
  sysplexname = ''
/>
```

inputcreatesystem.xml

```
<?xml version='1.0'?>
<!--=====================================================================-->
<!--   File name:  inputcreatesystem.xml                                 -->
<!--                                                                     -->
<!--   Descriptive name:  ...                                            -->
<!--                                                                     -->
<!--   Proprietary statement:                                            -->
<!--                                                                     -->
<!-- Licensed Material - Property of IBM                                 -->
<!--                                                                     -->
<!-- 5655-F31 (C) Copyright IBM Corp. 2000, 2001                         -->
<!-- All Rights Reserved.                                                -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or    -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.   -->
<!-- Status = H28W400                                                    -->
<!--                                                                     -->
<!--  Change history:                                                    -->
<!--$L0=OW44455, H28K510, 20001203, PDBL: Created.                       -->
```

```
<!--                                                                           -->
<!--===========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputcreatesystem [
<!ELEMENT inputcreatesystem EMPTY>
<!ATTLIST inputcreatesystem
  conversationname CDATA #REQUIRED
  systemname CDATA #REQUIRED
  systemdescription CDATA #IMPLIED
>
]>

<!--begin of default values-->
<inputcreatesystem
  conversationname = ''
  systemname = ''
  systemdescription = ''
/>
```

inputchangesystem.xml

```
<?xml version='1.0'?>
<!--===========================================================================-->
<!--   File name:  inputchangesystem.xml                                       -->
<!--                                                                           -->
<!--   Descriptive name:  ...                                                  -->
<!--                                                                           -->
<!--   Proprietary statement:                                                  -->
<!--                                                                           -->
<!-- Licensed Material - Property of IBM                                       -->
<!--                                                                           -->
<!-- 5655-F31 (C) Copyright IBM Corp. 2000, 2001                               -->
<!-- All Rights Reserved.                                                      -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or          -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.         -->
<!-- Status = H28W400                                                          -->
<!--                                                                           -->
<!--  Change history:                                                          -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                             -->
<!--$L1=OW44455, CB4.0, 20001205, PDBL: OLT support, security enhancement      -->
<!--                                                                           -->
<!--===========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputchangesystem [
<!ELEMENT inputchangesystem EMPTY>
<!ATTLIST inputchangesystem
  conversationname CDATA #REQUIRED
  systemname CDATA #REQUIRED
  systemdecription CDATA #IMPLIED
>
]>

<!--begin of default values-->
<inputchangesystem
```

```
  conversationname = ''
  systemname = ''
  systemdescription = ''
>
</inputchangesystem>
```

inputlistsystem.xml

```
<?xml version='1.0'?>
<!--==========================================================================-->
<!--   File name:  inputlistsystem.xml                                  -->
<!--                                                                         -->
<!--   Descriptive name:  ...                                                -->
<!--                                                                         -->
<!--   Proprietary statement:                                                -->
<!--                                                                         -->
<!-- Licensed Material - Property of IBM                                     -->
<!--                                                                         -->
<!-- 5655-F31 (C) Copyright IBM Corp. 2000, 2001                             -->
<!-- All Rights Reserved.                                                    -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or        -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.       -->
<!-- Status = H28W400                                                        -->
<!--                                                                         -->
<!--  Change history:                                                        -->
<!--$L0=OW44455, H28K510, 20001205, PDBL: Created.                           -->
<!--                                                                         -->
<!--==========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputlistsystem [
<!ELEMENT inputlistsystem EMPTY>
<!ATTLIST inputlistsystem
  conversationname CDATA #REQUIRED
  systemname CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputlistsystem
  conversationname = ''
  systemname = ''
/>
```

inputdeletesystem.xml

```
<?xml version='1.0'?>
<!--==========================================================================-->
<!--   File name:  inputdeletesystem.xml                                     -->
<!--                                                                         -->
<!--   Descriptive name:  ...                                                -->
<!--                                                                         -->
<!--   Proprietary statement:                                                -->
<!--                                                                         -->
<!-- Licensed Material - Property of IBM                                     -->
```

```
<!--                                                                        -->
<!-- 5655-F31 (C) Copyright IBM Corp. 2000, 2001                            -->
<!-- All Rights Reserved.                                                   -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or       -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.      -->
<!-- Status = H28W400                                                       -->
<!--                                                                        -->
<!--  Change history:                                                       -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                          -->
<!--                                                                        -->
<!--=======================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputdeletesystem [
<!ELEMENT inputdeletesystem EMPTY>
<!ATTLIST inputdeletesystem
  conversationname CDATA #REQUIRED
  systemname CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputdeletesystem
  conversationname = ''
  systemname = ''
/>
```

inputcreateserver.xml

```
<?xml version='1.0'?>
<!--=======================================================================-->
<!--   File name:  inputcreateserver.xml                                    -->
<!--                                                                        -->
<!--   Descriptive name:  ...                                               -->
<!--                                                                        -->
<!--   Proprietary statement:                                               -->
<!--                                                                        -->
<!-- Licensed Material - Property of IBM                                    -->
<!--                                                                        -->
<!-- 5655-F31 (C) Copyright IBM Corp. 2000, 2001                            -->
<!-- All Rights Reserved.                                                   -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or       -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.      -->
<!-- Status = H28W400                                                       -->
<!--                                                                        -->
<!--  Change history:                                                       -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                          -->
<!--$L1=OW44455, CB4.0, 20001205, PDBL: OLT support, security enhancement   -->
<!--$L2=OW44455, CB4.0, 20010124, PDBL: Kerberos, AssertedID support        -->
<!--                                                                        -->
<!--=======================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputcreateserver [
<!ELEMENT security EMPTY>
<!ATTLIST security
```

```
 value CDATA #REQUIRED
>
<!ELEMENT environment EMPTY>
<!ATTLIST environment
 value CDATA #REQUIRED
 name CDATA #REQUIRED
>
<!ELEMENT inputcreateserver (environment*,security*)>
<!ATTLIST inputcreateserver
  acceptassertedid (Y|N) #REQUIRED
  allowkerberos (Y|N) #REQUIRED
  allownonauthenticatedclients (Y|N) #REQUIRED
  allowserverregiongarbagecollection (Y|N) #REQUIRED
  allowuseridpasswd (Y|N) #REQUIRED
  allowssl (Y|N) #REQUIRED
  allowsslclientcerts (Y|N) #REQUIRED
  conversationname CDATA #REQUIRED
  dcekeytabfile  CDATA #REQUIRED
  dcequalityofprotection CDATA #REQUIRED
  debuggerallowed (Y|N) #REQUIRED
  garbagecollectioninterval CDATA #REQUIRED
  identityofthecontrolregion CDATA #REQUIRED
  identityoftheserverregion CDATA #REQUIRED
  isolationpolicy CDATA #REQUIRED
  localidentity CDATA #REQUIRED
  logstreamname CDATA #REQUIRED
  olthostname CDATA #REQUIRED
  oltport CDATA #REQUIRED
  procname CDATA #REQUIRED
  productionserver (Y|N) #REQUIRED
  remoteidentity CDATA #REQUIRED
  replicationpolicy CDATA #REQUIRED
  sendassertedid (Y|N) #REQUIRED
  serverdescription CDATA #IMPLIED
  servername CDATA #REQUIRED
  serverregionjvmname CDATA #REQUIRED
  serverregionrequiresjvm (Y|N) #REQUIRED
  serverregionstacksize CDATA #REQUIRED
  smfwrserveractivity (Y|N) #REQUIRED
  smfwrcontaineractivity (Y|N) #REQUIRED
  smfwrserverinterval (Y|N) #REQUIRED
  smfwrcontainerinterval (Y|N) #REQUIRED
  smfintervallength CDATA #REQUIRED
  sslracfkeyring CDATA #REQUIRED
  sslv2timeout CDATA #REQUIRED
  sslv3timeout CDATA #REQUIRED
  transactionfactory (Y|N) #REQUIRED
  usedce (Y|N) #REQUIRED
  useridpassticket (Y|N) #REQUIRED
>
]>

<!--begin of default values-->
<inputcreateserver
  acceptassertedid = 'N'
  allowkerberos = 'N'
  allownonauthenticatedclients = 'N'
  allowserverregiongarbagecollection = 'Y'
  allowuseridpasswd = 'N'
  allowssl = 'N'
  allowsslclientcerts = 'N'
  conversationname = ''
  dcekeytabfile = ''
  dcequalityofprotection = 'No_Protection'
  debuggerallowed = 'Y'
  garbagecollectioninterval = '50000'
  identityofthecontrolregion = ''
  identityoftheserverregion = ''
  isolationpolicy = 'One_Transaction_Per_Server_Region'
  localidentity = ''
  logstreamname = ''
  olthostname = ''
  oltport = '5000'
  procname = ''
  productionserver = 'N'
  remoteidentity = ''
  replicationpolicy = 'Replicate_As_Needed'
  sendassertedid = 'N'
  serverdescription = ''
  servername = ''
  serverregionjvmname = ''
  serverregionrequiresjvm = 'N'
  serverregionstacksize = ''
  smfwrserveractivity = 'N'
  smfwrcontaineractivity = 'N'
  smfwrserverinterval = 'N'
  smfwrcontainerinterval = 'N'
  smfintervallength = '3600'
  sslracfkeyring = 'CBKeyring'
  sslv2timeout = '100'
  sslv3timeout = '600'
  transactionfactory = 'N'
  usedce = 'N'
  useridpassticket = 'N'
>
</inputcreateserver>
```

inputdeleteserver.xml

```
<?xml version='1.0'?>
<!--=====================================================================-->
<!--   File name:  inputdeleteserver.xml                                 -->
<!--                                                                     -->
<!--   Descriptive name:  ...                                            -->
<!--                                                                     -->
<!--   Proprietary statement:                                            -->
<!--                                                                     -->
<!-- Licensed Material - Property of IBM                                 -->
<!--                                                                     -->
```

```
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                 -->
<!-- All Rights Reserved.                                                  -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or      -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.     -->
<!-- Status = H28K510                                                      -->
<!--                                                                       -->
<!--  Change history:                                                      -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                         -->
<!--                                                                       -->
<!--=====================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputdeleteserverinstance [
<!ELEMENT inputdeleteserverinstance EMPTY>
<!ATTLIST inputdeleteserverinstance
 conversationname CDATA #REQUIRED
 servername CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputdeleteserverinstance
 conversationname = ''
 servername = ''
/>
```

inputchangeserver.xml

```
<?xml version='1.0'?>
<!--=====================================================================-->
<!--   File name:  inputchangeserver.xml                                   -->
<!--                                                                       -->
<!--   Descriptive name:  ...                                              -->
<!--                                                                       -->
<!--   Proprietary statement:                                              -->
<!--                                                                       -->
<!-- Licensed Material - Property of IBM                                   -->
<!--                                                                       -->
<!-- 5655-F31 (C) Copyright IBM Corp. 2000, 2001                           -->
<!-- All Rights Reserved.                                                  -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or      -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.     -->
<!-- Status = H28W400                                                      -->
<!--                                                                       -->
<!--  Change history:                                                      -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                         -->
<!--$L1=OW44455, CB4.0, 20001205, PDBL: OLT support, security enhancement  -->
<!--$L2=OW44455, CB4.0, 20010124, PDBL: Kerberos, AssertedID support       -->
<!--                                                                       -->
<!--=====================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputchangeserver [
<!ELEMENT security EMPTY>
<!ATTLIST security
 value CDATA #REQUIRED
>
<!ELEMENT environment EMPTY><!ATTLIST environment
 value CDATA #REQUIRED
 name CDATA #REQUIRED
>
<!ELEMENT inputchangeserver (environment*,security*)>
<!ATTLIST inputchangeserver
  acceptassertedid CDATA #REQUIRED
  allowkerberos CDATA #REQUIRED
  allownonauthenticatedclients CDATA #REQUIRED
  allowserverregiongarbagecollection CDATA #REQUIRED
  allowuseridpasswd CDATA #REQUIRED
  allowssl CDATA #REQUIRED
  allowsslclientcerts CDATA #REQUIRED
  conversationname CDATA #REQUIRED
  dcekeytabfile  CDATA #REQUIRED
  dcequalityofprotection CDATA #REQUIRED
  debuggerallowed CDATA #REQUIRED
  garbagecollectioninterval CDATA #REQUIRED
  identityofthecontrolregion CDATA #REQUIRED
  identityoftheserverregion CDATA #REQUIRED
  isolationpolicy CDATA #REQUIRED
  localidentity CDATA #REQUIRED
  logstreamname CDATA #IMPLIED
  olthostname CDATA #IMPLIED
  oltport CDATA #REQUIRED
  procname CDATA #REQUIRED
  productionserver CDATA #REQUIRED
  remoteidentity CDATA #REQUIRED
  replicationpolicy CDATA #REQUIRED
  sendassertedid CDATA #REQUIRED
  serverdescription CDATA #IMPLIED
  servername CDATA #REQUIRED
  serverregionjvmname CDATA #IMPLIED
  serverregionrequiresjvm CDATA #REQUIRED
  serverregionstacksize CDATA #REQUIRED
  smfwrserveractivity CDATA #IMPLIED
  smfwrcontaineractivity CDATA #IMPLIED
  smfwrserverinterval CDATA #IMPLIED
  smfwrcontainerinterval CDATA #IMPLIED
  smfintervallength CDATA #IMPLIED
  sslracfkeyring CDATA #IMPLIED
  sslv2timeout CDATA #REQUIRED
  sslv3timeout CDATA #REQUIRED
  transactionfactory CDATA #REQUIRED
  usedce CDATA #REQUIRED
  useridpassticket CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputchangeserver
  acceptassertedid = ''
  allowkerberos = ''
  allownonauthenticatedclients = ''
  allowserverregiongarbagecollection = ''
  allowuseridpasswd = ''
  allowssl = ''
  allowsslclientcerts = ''
  conversationname = ''
  dcekeytabfile = ''
  dcequalityofprotection = ''
  debuggerallowed = ''
  garbagecollectioninterval = ''
  identityofthecontrolregion = ''
  identityoftheserverregion = ''
  isolationpolicy = ''
  localidentity = ''
  logstreamname = ''
  olthostname = ''
  oltport = ''
  procname = ''
  productionserver = ''
  remoteidentity = ''
  replicationpolicy = ''
  sendassertedid = ''
  serverdescription = ''
  servername = ''
  serverregionjvmname = ''
  serverregionrequiresjvm = ''
  serverregionstacksize = ''
  smfwrserveractivity = ''
  smfwrcontaineractivity = ''
  smfwrserverinterval = ''
  smfwrcontainerinterval = ''
  smfintervallength = ''
  sslracfkeyring = ''
  sslv2timeout = ''
  sslv3timeout = ''
  transactionfactory = ''
  usedce = ''
  useridpassticket = ''
>
</inputchangeserver>
```

inputlistserver.xml

```
<?xml version='1.0'?>
<!--==================================================================-->
<!--   File name:  inputlistserver.xml                                 -->
<!--                                                                   -->
<!--   Descriptive name:  ...                                          -->
<!--                                                                   -->
<!--   Proprietary statement:                                          -->
<!--                                                                   -->
<!-- Licensed Material - Property of IBM                               -->
<!--                                                                   -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                             -->
<!-- All Rights Reserved.                                              -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or  -->
```

```
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.      -->
<!-- Status = H28K510                                                       -->
<!--                                                                        -->
<!--  Change history:                                                       -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                          -->
<!--                                                                        -->
<!--======================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputlistserver [
<!ELEMENT inputlistserver EMPTY>
<!ATTLIST inputlistserver
 conversationname CDATA #REQUIRED
 servername CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputlistserver
 conversationname = ''
 servername = ''
/>
```

inputlistj2eeapplication.xml

```
<?xml version='1.0'?>
<!--======================================================================-->
<!--   File name:  inputlistj2eeapplication.xml                             -->
<!--                                                                        -->
<!--   Descriptive name:  ...                                               -->
<!--                                                                        -->
<!--   Proprietary statement:                                               -->
<!--                                                                        -->
<!-- Licensed Material - Property of IBM                                    -->
<!--                                                                        -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                  -->
<!-- All Rights Reserved.                                                   -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or       -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.      -->
<!-- Status = H28K510                                                       -->
<!--                                                                        -->
<!--  Change history:                                                       -->
<!--$L0=OW44455, H28K510, 20010125, PDBL: Created.                          -->
<!--                                                                        -->
<!--======================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputlistj2eeapplication [
<!ELEMENT inputlistj2eeapplication EMPTY>
<!ATTLIST inputlistj2eeapplication
  conversationname CDATA #REQUIRED
  j2eeservername CDATA #REQUIRED
  j2eeapplicationname CDATA #REQUIRED
>
]>

<!--begin of default values-->
```

```
<inputlistj2eeapplication
  conversationname = ''
  j2eeservername = ''
  j2eeapplicationname = ''
/>
```

inputdeletej2eeapplication.xml

```
<?xml version='1.0'?>
<!--=========================================================================-->
<!--   File name:  inputdeletej2eeapplication.xml                           -->
<!--                                                                         -->
<!--   Descriptive name:  ...                                                -->
<!--                                                                         -->
<!--   Proprietary statement:                                                -->
<!--                                                                         -->
<!-- Licensed Material - Property of IBM                                     -->
<!--                                                                         -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2001                                   -->
<!-- All Rights Reserved.                                                    -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or       -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.      -->
<!-- Status = H28K510                                                        -->
<!--                                                                         -->
<!--  Change history:                                                        -->
<!--$L0=OW44455, CB4.0, 20000124, PDBL: Created                              -->
<!--                                                                         -->
<!--=========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputdeletej2eeapplication [
<!ELEMENT inputdeletej2eeapplication EMPTY>
<!ATTLIST inputdeletej2eeapplication
  conversationname CDATA #REQUIRED
  j2eeservername CDATA #REQUIRED
  j2eeapplicationname CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputdeletej2eeapplication
  conversationname = ''
  j2eeservername = ''
  j2eeapplicationname = ''
/>
```

inputlistj2eecomponents.xml

```
<?xml version='1.0'?>
<!--=========================================================================-->
<!--   File name:  inputlistj2eecomponents.xml                               -->
<!--                                                                         -->
<!--   Descriptive name:  ...                                                -->
<!--                                                                         -->
<!--   Proprietary statement:                                                -->
```

```
<!--                                                                        -->
<!-- Licensed Material - Property of IBM                                    -->
<!--                                                                        -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                  -->
<!-- All Rights Reserved.                                                   -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or       -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.      -->
<!-- Status = H28K510                                                       -->
<!--                                                                        -->
<!--  Change history:                                                       -->
<!--$L0=OW44455, H28K510, 20010125, PDBL: Created.                          -->
<!--                                                                        -->
<!--========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputlistj2eecomponents [
<!ELEMENT inputlistj2eecomponents EMPTY>
<!ATTLIST inputlistj2eecomponents
  conversationname CDATA #REQUIRED
  j2eeservername CDATA #REQUIRED
  j2eeapplicationname CDATA #REQUIRED
  modulename CDATA #REQUIRED
  componentname CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputlistj2eecomponents
  conversationname = ''
  j2eeservername = ''
  j2eeapplicationname = ''
  modulename = ''
  componentname = ''
/>
```

inputlistj2eemodules.xml

```
<?xml version='1.0'?>
<!--========================================================================-->
<!--   File name:  inputlistj2eemodules.xml                                 -->
<!--                                                                        -->
<!--   Descriptive name:  ...                                               -->
<!--                                                                        -->
<!--   Proprietary statement:                                               -->
<!--                                                                        -->
<!--   Licensed Material - Property of IBM                                  -->
<!--   5655-F31 (C) Copyright IBM Corp. 2000, 2001                          -->
<!--                                                                        -->
<!--   All Rights Reserved.                                                 -->
<!--                                                                        -->
<!--   U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or     -->
<!--   Disclosure restricted by GSA-ADP schedule contract with IBM Corp.    -->
<!--                                                                        -->
<!--   Status = H28W400                                                     -->
```

```
<!--                                                                        -->
<!--  Change history:                                                       -->
<!--$L0=OW44455, H28K510, 20010125, PDBL: Created.                          -->
<!--                                                                        -->
<!--========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputlistj2eemodules [
<!ELEMENT inputlistj2eemodules EMPTY>
<!ATTLIST inputlistj2eemodules
  conversationname CDATA #REQUIRED
  j2eeservername CDATA #REQUIRED
  j2eeapplicationname CDATA #REQUIRED
  modulename CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputlistj2eemodules
  conversationname = ''
  j2eeservername = ''
  j2eeapplicationname = ''
  modulename = ''
/>
```

inputcreatej2eeserver.xml

```
<?xml version='1.0'?>
<!--========================================================================-->
<!--   File name:  inputcreatej2eeserver.xml                                -->
<!--                                                                        -->
<!--   Descriptive name:  ...                                               -->
<!--                                                                        -->
<!--   Proprietary statement:                                               -->
<!--                                                                        -->
<!--   Licensed Material - Property of IBM                                  -->
<!--   5655-F31 (C) Copyright IBM Corp. 2000, 2001                          -->
<!--                                                                        -->
<!--   All Rights Reserved.                                                 -->
<!--                                                                        -->
<!--   U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or     -->
<!--   Disclosure restricted by GSA-ADP schedule contract with IBM Corp.    -->
<!--                                                                        -->
<!--   Status = H28W400                                                     -->
<!--                                                                        -->
<!--  Change history:                                                       -->
<!--$L0=OW44455, H28K510, 20000721, PDBL: Created.                          -->
<!--$L1=OW44455, CB4.0, 20001205, PDBL: OLT support, security enhancement   -->
<!--$L2=OW44455, CB4.0, 20010125, PDBL: kerberos, assertedID support        -->
<!--                                                                        -->
<!--========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputcreatej2eeserver [
<!ELEMENT security EMPTY>
<!ATTLIST security
```

```
 value CDATA #REQUIRED
>
<!ELEMENT environment EMPTY>
<!ATTLIST environment
 value CDATA #REQUIRED
 name CDATA #REQUIRED
>
<!ELEMENT inputcreatej2eeserver (environment*,security*)>
<!ATTLIST inputcreatej2eeserver
  acceptassertedid (Y|N) #REQUIRED
  allowkerberos (Y|N) #REQUIRED
  allownonauthenticatedclients (Y|N) #REQUIRED
  allowserverregiongarbagecollection (Y|N) #REQUIRED
  allowuseridpasswd (Y|N) #REQUIRED
  allowssl (Y|N) #REQUIRED
  allowsslclientcerts (Y|N) #REQUIRED
  conversationname CDATA #REQUIRED
  dcekeytabfile  CDATA #REQUIRED
  dcequalityofprotection CDATA #REQUIRED
  debuggerallowed (Y|N) #REQUIRED
  garbagecollectioninterval CDATA #REQUIRED
  identityofthecontrolregion CDATA #REQUIRED
  identityoftheserverregion CDATA #REQUIRED
  isolationpolicy CDATA #REQUIRED
  j2eeserverdescription CDATA #IMPLIED
  j2eeservername CDATA #REQUIRED
  localidentity CDATA #REQUIRED
  logstreamname CDATA #REQUIRED
  olthostname CDATA #REQUIRED
  oltport CDATA #REQUIRED
  procname CDATA #REQUIRED
  productionserver (Y|N) #REQUIRED
  remoteidentity CDATA #REQUIRED
  replicationpolicy CDATA #REQUIRED
  sendassertedid (Y|N) #REQUIRED
  serverregionjvmname CDATA #REQUIRED
  serverregionrequiresjvm (Y|N) #REQUIRED
  serverregionstacksize CDATA #REQUIRED
  sslracfkeyring CDATA #REQUIRED
  sslv2timeout CDATA #REQUIRED
  sslv3timeout CDATA #REQUIRED
  transactionfactory (Y|N) #REQUIRED
  usedce (Y|N) #REQUIRED
  useridpassticket (Y|N) #REQUIRED
>
]>

<!--begin of default values-->
<inputcreatej2eeserver
  acceptassertedid = 'N'
  allowkerberos = 'N'
  allownonauthenticatedclients = 'N'
  allowserverregiongarbagecollection = 'Y'
  allowuseridpasswd = 'N'
  allowssl = 'N'
  allowsslclientcerts = 'N'
  conversationname = ''
  dcekeytabfile = ''
  dcequalityofprotection = 'No_Protection'
  debuggerallowed = 'Y'
  garbagecollectioninterval = '50000'
  identityofthecontrolregion = ''
  identityoftheserverregion = ''
  isolationpolicy = 'One_Transaction_Per_Server_Region'
  j2eeserverdescription = ''
  j2eeservername = ''
  localidentity = ''
  logstreamname = ''
  olthostname = ''
  oltport = '5000'
  procname = ''
  productionserver = 'N'
  remoteidentity = ''
  replicationpolicy = 'Replicate_As_Needed'
  sendassertedid = 'N'
  serverregionjvmname = ''
  serverregionrequiresjvm = 'N'
  serverregionstacksize = ''
  sslracfkeyring = 'CBKeyring'
  sslv2timeout = '100'
  sslv3timeout = '600'
  transactionfactory = 'N'
  usedce = 'N'
  useridpassticket = 'N'
>
</inputcreatej2eeserver>
```

inputdeletej2eeserver.xml

```
<?xml version='1.0'?>
<!--==================================================================-->
<!--   File name:  inputdeletej2eeserver.xml                          -->
<!--                                                                  -->
<!--   Descriptive name:  ...                                         -->
<!--                                                                  -->
<!--   Proprietary statement:                                         -->
<!--                                                                  -->
<!-- Licensed Material - Property of IBM                              -->
<!--                                                                  -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2001                            -->
<!-- All Rights Reserved.                                             -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp. -->
<!-- Status = H28K510                                                 -->
<!--                                                                  -->
<!--  Change history:                                                 -->
<!--$L0=OW44455, CB4.0, 20000124, PDBL: Created                       -->
<!--                                                                  -->
<!--==================================================================-->
```

```
<!-- internal DTD -->
<!DOCTYPE inputdeletej2eeserver [
<!ELEMENT inputdeletej2eeserver EMPTY>
<!ATTLIST inputdeletej2eeserver
 conversationname CDATA #REQUIRED
 j2eeservername CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputj2eedeleteserver
 conversationname = ''
 j2eeservername = ''
/>
```

inputchangej2eeserver.xml

```
<?xml version='1.0'?>
<!--=====================================================================-->
<!--   File name:  inputchangej2eeserver.xml                             -->
<!--                                                                     -->
<!--   Descriptive name:  ...                                            -->
<!--                                                                     -->
<!--   Proprietary statement:                                            -->
<!--                                                                     -->
<!--                                                                     -->
<!--   Licensed Material - Property of IBM                               -->
<!--   5655-F31 (C) Copyright IBM Corp. 2000, 2001                       -->
<!--                                                                     -->
<!--   All Rights Reserved.                                              -->
<!--                                                                     -->
<!--   U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or  -->
<!--   Disclosure restricted by GSA-ADP schedule contract with IBM Corp. -->
<!--                                                                     -->
<!--   Status = H28W400                                                  -->
<!--                                                                     -->
<!--  Change history:                                                    -->
<!--$L0=OW44455, H28K510, 20000721, PDBL: Created.                       -->
<!--$L1=OW44455, CB4.0, 20001205, PDBL: OLT support, security enhancement -->
<!--$L2=OW44455, CB4.0, 20010125, PDBL: kerberos, assertedID support     -->
<!--                                                                     -->
<!--=====================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputchangej2eeserver [
<!ELEMENT security EMPTY>
<!ATTLIST security
 value CDATA #REQUIRED
>
<!ELEMENT environment EMPTY><!ATTLIST environment
 value CDATA #REQUIRED
 name CDATA #REQUIRED
>
<!ELEMENT inputchangej2eeserver (environment*,security*)>
<!ATTLIST inputchangej2eeserver
```

```
  acceptassertedid CDATA #REQUIRED
  allowkerberos CDATA #REQUIRED
  allownonauthenticatedclients CDATA #REQUIRED
  allowserverregiongarbagecollection CDATA #REQUIRED
  allowuseridpasswd CDATA #REQUIRED
  allowssl CDATA #REQUIRED
  allowsslclientcerts CDATA #REQUIRED
  conversationname CDATA #REQUIRED
  dcekeytabfile  CDATA #REQUIRED
  dcequalityofprotection CDATA #REQUIRED
  debuggerallowed CDATA #REQUIRED
  garbagecollectioninterval CDATA #REQUIRED
  identityofthecontrolregion CDATA #REQUIRED
  identityoftheserverregion CDATA #REQUIRED
  isolationpolicy CDATA #REQUIRED
  j2eeserverdescription CDATA #IMPLIED
  j2eeservername CDATA #REQUIRED
  localidentity CDATA #REQUIRED
  logstreamname CDATA #IMPLIED
  olthostname CDATA #IMPLIED
  oltport CDATA #REQUIRED
  procname CDATA #REQUIRED
  productionserver CDATA #REQUIRED
  remoteidentity CDATA #REQUIRED
  replicationpolicy CDATA #REQUIRED
  sendassertedid CDATA #REQUIRED
  serverregionjvmname CDATA #IMPLIED
  serverregionrequiresjvm CDATA #REQUIRED
  serverregionstacksize CDATA #REQUIRED
  sslracfkeyring CDATA #IMPLIED
  sslv2timeout CDATA #REQUIRED
  sslv3timeout CDATA #REQUIRED
  transactionfactory CDATA #REQUIRED
  usedce CDATA #REQUIRED
  useridpassticket CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputchangej2eeserver
  acceptassertedid = ''
  allowkerberos = ''
  allownonauthenticatedclients = ''
  allowserverregiongarbagecollection = ''
  allowuseridpasswd = ''
  allowssl = ''
  allowsslclientcerts = ''
  conversationname = ''
  dcekeytabfile = ''
  dcequalityofprotection = ''
  debuggerallowed = ''
  garbagecollectioninterval = ''
  identityofthecontrolregion = ''
  identityoftheserverregion = ''
  isolationpolicy = ''
  j2eeserverdescription = ''
  j2eeservername = ''
  localidentity = ''
  logstreamname = ''
  olthostname = ''
  oltport = ''
  procname = ''
  productionserver = ''
  remoteidentity = ''
  replicationpolicy = ''
  sendassertedid = ''
  serverregionjvmname = ''
  serverregionrequiresjvm = ''
  serverregionstacksize = ''
  sslracfkeyring = ''
  sslv2timeout = ''
  sslv3timeout = ''
  transactionfactory = ''
  usedce = ''
  useridpassticket = ''
>
</inputchangej2eeserver>
```

inputlistj2eeserver.xml

```
<?xml version='1.0'?>
<!--=========================================================================-->
<!--   File name:  inputlistj2eeserver.xml                                   -->
<!--                                                                         -->
<!--   Descriptive name:  ...                                                -->
<!--                                                                         -->
<!--   Proprietary statement:                                                -->
<!--                                                                         -->
<!-- Licensed Material - Property of IBM                                     -->
<!--                                                                         -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2001                                   -->
<!-- All Rights Reserved.                                                    -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or        -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.       -->
<!-- Status = H28K510                                                        -->
<!--                                                                         -->
<!--  Change history:                                                        -->
<!--$L0=OW44455, CB4.0, 20000124, PDBL: Created                              -->
<!--                                                                         -->
<!--=========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputlistj2eeserver [
<!ELEMENT inputlistj2eeserver EMPTY>
<!ATTLIST inputlistj2eeserver
 conversationname CDATA #REQUIRED
 j2eeservername CDATA #REQUIRED
>
]>
```

```
<!--begin of default values-->
<inputlistj2eeserver
 conversationname = ''
 j2eeservername = ''
/>
```

inputcreateserverinstance.xml

```
<?xml version='1.0'?>
<!--=====================================================================-->
<!--   File name:  inputcreateserverinstance.xml                         -->
<!--                                                                     -->
<!--   Descriptive name:  ...                                            -->
<!--                                                                     -->
<!--   Proprietary statement:                                            -->
<!--                                                                     -->
<!-- Licensed Material - Property of IBM                                 -->
<!--                                                                     -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000, 2001                         -->
<!-- All Rights Reserved.                                                -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or    -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.   -->
<!-- Status = H28K510                                                    -->
<!--                                                                     -->
<!--  Change history:                                                    -->
<!--$L0=OW44455, H28K510, 20000721, PDBL: Created.                       -->
<!--$L1=OW44455, CB4.0, 20010124, PDBL: attribute configportnumber added -->
<!--$L2=OW44455, CB4.x, 20010226, PDBL: attribute sslfirewallport added  -->
<!--                                                                     -->
<!--=====================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputcreateserverinstance [
<!ELEMENT environment EMPTY>
<!ATTLIST environment
 value CDATA #REQUIRED
 name CDATA #REQUIRED
>
<!ELEMENT inputcreateserverinstance (environment*)>
<!ATTLIST inputcreateserverinstance
 conversationname CDATA #REQUIRED
 servername CDATA #REQUIRED
 serverinstancename CDATA #REQUIRED
 serverinstancedescription CDATA #IMPLIED
 systemname CDATA #REQUIRED
 logstreamname CDATA #IMPLIED
 configportnumber CDATA #REQUIRED
 sslfirewallport CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputcreateserverinstance
 conversationname = ''
 servername = ''
 serverinstancename = ''
 serverinstancedescription = ''
 systemname = 'SY1'
 logstreamname = ''
 configportnumber = '9000'
 sslfirewallport = '0'
>
</inputcreateserverinstance>
```

inputdeleteserverinstance.xml

```
<?xml version='1.0'?>
<!--=========================================================================-->
<!--   File name:  inputdeleteserverinstance.xml                            -->
<!--                                                                         -->
<!--   Descriptive name:  ...                                                -->
<!--                                                                         -->
<!--   Proprietary statement:                                                -->
<!--                                                                         -->
<!-- Licensed Material - Property of IBM                                     -->
<!--                                                                         -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                   -->
<!-- All Rights Reserved.                                                    -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or       -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.      -->
<!-- Status = H28K510                                                        -->
<!--                                                                         -->
<!--  Change history:                                                        -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                          -->
<!--                                                                         -->
<!--=========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputdeleteserverinstance [
<!ELEMENT inputdeleteserverinstance EMPTY>
<!ATTLIST inputdeleteserverinstance
 conversationname CDATA #REQUIRED
 servername CDATA #REQUIRED
 serverinstancename CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputdeleteserverinstance
 conversationname = ''
 servername = ''
 serverinstancename = ''
/>
```

inputchangeserverinstance.xml

```
<?xml version='1.0'?>
<!--=========================================================================-->
<!--   File name:  inputchangeserverinstance.xml                            -->
<!--                                                                         -->
```

```
<!--   Descriptive name:  ...                                              -->
<!--                                                                       -->
<!--   Proprietary statement:                                              -->
<!--                                                                       -->
<!-- Licensed Material - Property of IBM                                   -->
<!--                                                                       -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000, 2001                           -->
<!-- All Rights Reserved.                                                  -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or      -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.     -->
<!-- Status = H28K510                                                      -->
<!--                                                                       -->
<!--  Change history:                                                      -->
<!--$L0=OW44455, H28K510, 20000721, PDBL: Created.                         -->
<!--$L1=OW44455, CB4.0, 20010124, PDBL: attribute configportnumber added   -->
<!--$L2=OW44455, CB4.x, 20010226, PDBL: attribute sslfirewallport added    -->
<!--                                                                       -->
<!--=======================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputchangeserverinstance [
<!ELEMENT environment EMPTY>
<!ATTLIST environment
 value CDATA #REQUIRED
 name CDATA #REQUIRED
>
<!ELEMENT inputchangeserverinstance (environment*)>
<!ATTLIST inputchangeserverinstance
 conversationname CDATA #REQUIRED
 servername CDATA #REQUIRED
 serverinstancename CDATA #REQUIRED
 serverinstancedescription CDATA #IMPLIED
 systemname CDATA #REQUIRED
 logstreamname CDATA #IMPLIED
 configportnumber CDATA #REQUIRED
 sslfirewallport CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputchangeserverinstance
 conversationname = ''
 servername = ''
 serverinstancename = ''
 serverinstancedescription = ''
 systemname = ''
 logstreamname = ''
 configportnumber = ''
 sslfirewallport = ''
>
</inputchangeserverinstance>
```

inputlistserverinstance.xml

```
<?xml version='1.0'?>
<!--===================================================================-->
<!--   File name:  inputlistserverinstance.xml                          -->
<!--                                                                     -->
<!--   Descriptive name:  ...                                            -->
<!--                                                                     -->
<!--   Proprietary statement:                                            -->
<!--                                                                     -->
<!-- Licensed Material - Property of IBM                                 -->
<!--                                                                     -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                               -->
<!-- All Rights Reserved.                                                -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or    -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.   -->
<!-- Status = H28K510                                                    -->
<!--                                                                     -->
<!--  Change history:                                                    -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                       -->
<!--                                                                     -->
<!--===================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputlistserverinstance [
<!ELEMENT inputlistserverinstance EMPTY>
<!ATTLIST inputlistserverinstance
 conversationname CDATA #REQUIRED
 servername CDATA #REQUIRED
 serverinstancename CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputlistserverinstance
 conversationname = ''
 servername = ''
 serverinstancename = ''
/>
```

inputcreatecontainer.xml

```
<?xml version='1.0'?>
<!--===================================================================-->
<!--   File name:  inputcreatecontainer.xml                             -->
<!--                                                                     -->
<!--   Descriptive name:  ...                                            -->
<!--                                                                     -->
<!--   Proprietary statement:                                            -->
<!--                                                                     -->
<!-- Licensed Material - Property of IBM                                 -->
<!--                                                                     -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                               -->
<!-- All Rights Reserved.                                                -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or    -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.   -->
```

```
<!-- Status = H28K510                                                      -->
<!--                                                                       -->
<!--  Change history:                                                      -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                         -->
<!--                                                                       -->
<!--=====================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputcreatecontainer [
<!ELEMENT inputcreatecontainer EMPTY>
<!ATTLIST inputcreatecontainer
  conversationname CDATA #REQUIRED
  servername CDATA #REQUIRED
  containername CDATA #REQUIRED
  containerdescription CDATA #IMPLIED
  aclcheckrequired (Y|N) #REQUIRED
  activationisolationpolicy CDATA #REQUIRED
  passivationconstraints CDATA #REQUIRED
  managedobjectrefreshpolicy CDATA #REQUIRED
  transactionpolicy CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputcreatecontainer
  conversationname = ''
  servername = ''
  containername = ''
  containerdescription = ''
  aclcheckrequired = 'N'
  activationisolationpolicy = 'Transaction_Level'
  passivationconstraints = 'Not_Pinned'
  managedobjectrefreshpolicy = 'At_Activation'
  transactionpolicy = 'TX_Requires'
/>
```

inputdeletecontainer.xml

```
<?xml version='1.0'?>
<!--=====================================================================-->
<!--   File name:  inputdeletecontainer.xml                                -->
<!--                                                                       -->
<!--   Descriptive name:  ...                                              -->
<!--                                                                       -->
<!--   Proprietary statement:                                              -->
<!--                                                                       -->
<!-- Licensed Material - Property of IBM                                   -->
<!--                                                                       -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                 -->
<!-- All Rights Reserved.                                                  -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or      -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.     -->
<!-- Status = H28K510                                                      -->
<!--                                                                       -->
<!--  Change history:                                                      -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                         -->
```

```
<!--                                                                          -->
<!--==========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputdeletecontainer [
<!ELEMENT inputdeletecontainer EMPTY>
<!ATTLIST inputdeletecontainer
  conversationname CDATA #REQUIRED
  servername CDATA #REQUIRED
  containername CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputdeletecontainer
  conversationname = ''
  servername = ''
  containername = ''
/>
```

inputchangecontainer.xml

```
<?xml version='1.0'?>
<!--==========================================================================-->
<!--   File name:  inputchangecontainer.xml                                   -->
<!--                                                                          -->
<!--   Descriptive name:  ...                                                 -->
<!--                                                                          -->
<!--   Proprietary statement:                                                 -->
<!--                                                                          -->
<!-- Licensed Material - Property of IBM                                      -->
<!--                                                                          -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                    -->
<!-- All Rights Reserved.                                                     -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or         -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.        -->
<!-- Status = H28K510                                                         -->
<!--                                                                          -->
<!--  Change history:                                                         -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                            -->
<!--                                                                          -->
<!--==========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputchangecontainer [
<!ELEMENT inputchangecontainer EMPTY>
<!ATTLIST inputchangecontainer
  conversationname CDATA #REQUIRED
  servername CDATA #REQUIRED
  containername CDATA #REQUIRED
  containerdescription CDATA #IMPLIED
  aclcheckrequired CDATA #REQUIRED
  activationisolationpolicy CDATA #REQUIRED
  passivationconstraints CDATA #REQUIRED
  managedobjectrefreshpolicy CDATA #REQUIRED
  transactionpolicy CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputchangecontainer
  conversationname = ''
  servername = ''
  containername = ''
  containerdescription = ''
  aclcheckrequired = ''
  activationisolationpolicy = ''
  passivationconstraints = ''
  managedobjectrefreshpolicy = ''
  transactionpolicy = ''
/>
```

inputlistcontainer.xml

```
<?xml version='1.0'?>
<!--==========================================================================-->
<!--   File name:  inputlistcontainer.xml                                     -->
<!--                                                                          -->
<!--   Descriptive name:  ...                                                 -->
<!--                                                                          -->
<!--   Proprietary statement:                                                 -->
<!--                                                                          -->
<!-- Licensed Material - Property of IBM                                      -->
<!--                                                                          -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                    -->
<!-- All Rights Reserved.                                                     -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or         -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.        -->
<!-- Status = H28K510                                                         -->
<!--                                                                          -->
<!--  Change history:                                                         -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                            -->
<!--                                                                          -->
<!--==========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputlistcontainer [
<!ELEMENT inputlistcontainer EMPTY>
<!ATTLIST inputlistcontainer
  conversationname CDATA #REQUIRED
  servername CDATA #REQUIRED
  containername CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputlistcontainer
  conversationname = ''
  servername = ''
  containername = ''
/>
```

inputcreatelrm.xml

```
<?xml version='1.0'?>
<!--=====================================================================-->
<!--   File name:  inputcreatelrm.xml                                     -->
<!--                                                                      -->
<!--   Descriptive name:  ...                                             -->
<!--                                                                      -->
<!--   Proprietary statement:                                             -->
<!--                                                                      -->
<!-- Licensed Material - Property of IBM                                  -->
<!--                                                                      -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                -->
<!-- All Rights Reserved.                                                 -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or     -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.    -->
<!-- Status = H28K510                                                     -->
<!--                                                                      -->
<!--  Change history:                                                     -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                        -->
<!--                                                                      -->
<!--=====================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputcreatelrm [
<!ELEMENT inputcreatelrm EMPTY>
<!ATTLIST inputcreatelrm
 conversationname CDATA #REQUIRED
 lrmname CDATA #REQUIRED
 lrmdescription CDATA #IMPLIED
 coclassname CDATA #REQUIRED
 codllname CDATA #REQUIRED
 coclasscreatefunction CDATA #REQUIRED
 lrmsubsystemtype CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputcreatelrm
 conversationname = ''
 lrmname = ''
 lrmdescription = ''
 coclassname = ''
 codllname = ''
 coclasscreatefunction = ''
 lrmsubsystemtype = 'Generic'
/>
```

inputdeletelrm.xml

```
<?xml version='1.0'?>
<!--=====================================================================-->
<!--   File name:  inputdeletelrm.xml                                     -->
<!--                                                                      -->
<!--   Descriptive name:  ...                                             -->
<!--                                                                      -->
```

```
<!--   Proprietary statement:                                          -->
<!--                                                                   -->
<!-- Licensed Material - Property of IBM                               -->
<!--                                                                   -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                             -->
<!-- All Rights Reserved.                                              -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or  -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp. -->
<!-- Status = H28K510                                                  -->
<!--                                                                   -->
<!--  Change history:                                                  -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                     -->
<!--                                                                   -->
<!--=================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputdeletelrm [
<!ELEMENT inputdeletelrm EMPTY>
<!ATTLIST inputdeletelrm
 conversationname CDATA #REQUIRED
 lrmname CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputdeletelrm
 conversationname = ''
 lrmname = ''
/>
```

inputchangelrm.xml

```
<?xml version='1.0'?>
<!--=================================================================-->
<!--   File name:  inputchangelrm.xml                                  -->
<!--                                                                   -->
<!--   Descriptive name:  ...                                          -->
<!--                                                                   -->
<!--   Proprietary statement:                                          -->
<!--                                                                   -->
<!-- Licensed Material - Property of IBM                               -->
<!--                                                                   -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                             -->
<!-- All Rights Reserved.                                              -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or  -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp. -->
<!-- Status = H28K510                                                  -->
<!--                                                                   -->
<!--  Change history:                                                  -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                     -->
<!--                                                                   -->
<!--=================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputchangelrm [
<!ELEMENT inputchangelrm EMPTY>
<!ATTLIST inputchangelrm
```

```
 conversationname CDATA #REQUIRED
 lrmname CDATA #REQUIRED
 lrmdescription CDATA #IMPLIED
 coclassname CDATA #REQUIRED
 codllname CDATA #REQUIRED
 coclasscreatefunction CDATA #REQUIRED
 lrmsubsystemtype CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputchangelrm
 conversationname = ''
 lrmname = ''
 lrmdescription = ''
 coclassname = ''
 codllname = ''
 coclasscreatefunction = ''
 lrmsubsystemtype = ''
/>
```

inputlistlrm.xml

```
<?xml version='1.0'?>
<!--=====================================================================-->
<!--   File name:  inputlistlrm.xml                                      -->
<!--                                                                     -->
<!--   Descriptive name:  ...                                            -->
<!--                                                                     -->
<!--   Proprietary statement:                                            -->
<!--                                                                     -->
<!-- Licensed Material - Property of IBM                                 -->
<!--                                                                     -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                               -->
<!-- All Rights Reserved.                                                -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or    -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.   -->
<!-- Status = H28K510                                                    -->
<!--                                                                     -->
<!--  Change history:                                                    -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                       -->
<!--                                                                     -->
<!--=====================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputlistlrm [
<!ELEMENT inputlistlrm EMPTY>
<!ATTLIST inputlistlrm
 conversationname CDATA #REQUIRED
 lrmname CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputlistlrm
```

```
 conversationname = ''
 lrmname = ''
/>
```

inputcreatelrmi.xml

```
<?xml version='1.0'?>
<!--==========================================================================-->
<!--   File name:  inputcreatelrmi.xml                                        -->
<!--                                                                          -->
<!--   Descriptive name:  ...                                                 -->
<!--                                                                          -->
<!--   Proprietary statement:                                                 -->
<!--                                                                          -->
<!-- Licensed Material - Property of IBM                                      -->
<!--                                                                          -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                    -->
<!-- All Rights Reserved.                                                     -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or         -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.        -->
<!-- Status = H28K510                                                         -->
<!--                                                                          -->
<!--  Change history:                                                         -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                            -->
<!--                                                                          -->
<!--==========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputcreatelrmi [
<!ELEMENT connection EMPTY>
<!ATTLIST  connection
 value CDATA #REQUIRED
 name CDATA #REQUIRED
>
<!ELEMENT inputcreatelrmi (connection*)>
<!ATTLIST inputcreatelrmi
  conversationname CDATA #REQUIRED
  lrmname CDATA #REQUIRED
  lrminame CDATA #REQUIRED
  lrmidescription CDATA #IMPLIED
  systemname CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputcreatelrmi
  conversationname = ''
  lrmname = ''
  lrminame = ''
  lrmidescription = ''
  systemname = 'SY1'
>
</inputcreatelrmi>
```

inputdeletelrmi.xml

```
<?xml version='1.0'?>
<!--=========================================================================-->
<!--   File name:  inputdeletelrmi.xml                                        -->
<!--                                                                          -->
<!--   Descriptive name:  ...                                                 -->
<!--                                                                          -->
<!--   Proprietary statement:                                                 -->
<!--                                                                          -->
<!-- Licensed Material - Property of IBM                                      -->
<!--                                                                          -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                    -->
<!-- All Rights Reserved.                                                     -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or       -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.      -->
<!-- Status = H28K510                                                         -->
<!--                                                                          -->
<!--  Change history:                                                         -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                            -->
<!--                                                                          -->
<!--=========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputdeletelrmi [
<!ELEMENT inputdeletelrmi EMPTY>
<!ATTLIST inputdeletelrmi
 conversationname CDATA #REQUIRED
 lrmname CDATA #REQUIRED
 lrminame CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputdeletelrmi
 conversationname = ''
 lrmname = ''
 lrminame = ''
/>
```

inputchangelrmi.xml

```
<?xml version='1.0'?>
<!--=========================================================================-->
<!--   File name:  inputchangelrmi.xml                                        -->
<!--                                                                          -->
<!--   Descriptive name:  ...                                                 -->
<!--                                                                          -->
<!--   Proprietary statement:                                                 -->
<!--                                                                          -->
<!-- Licensed Material - Property of IBM                                      -->
<!--                                                                          -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                    -->
<!-- All Rights Reserved.                                                     -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or       -->
```

```
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.      -->
<!-- Status = H28K510                                                        -->
<!--                                                                         -->
<!--  Change history:                                                        -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                           -->
<!--                                                                         -->
<!--=========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputchangelrmi [
<!ELEMENT connection EMPTY>
<!ATTLIST  connection
 value CDATA #REQUIRED
 name CDATA #REQUIRED
>
<!ELEMENT inputchangelrmi (connection*)>
<!ATTLIST inputchangelrmi
  conversationname CDATA #REQUIRED
  lrmname CDATA #REQUIRED
  lrminame CDATA #REQUIRED
  lrmidescription CDATA #IMPLIED
  systemname CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputchangelrmi
  conversationname = ''
  lrmname = ''
  lrminame = ''
  lrmidescription = ''
  systemname = ''
>
</inputchangelrmi>
```

inputlistlrmi.xml

```
<?xml version='1.0'?>
<!--=========================================================================-->
<!--   File name:  inputlistlrmi.xml                                         -->
<!--                                                                         -->
<!--   Descriptive name:  ...                                                -->
<!--                                                                         -->
<!--   Proprietary statement:                                                -->
<!--                                                                         -->
<!-- Licensed Material - Property of IBM                                     -->
<!--                                                                         -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                   -->
<!-- All Rights Reserved.                                                    -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or       -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.      -->
<!-- Status = H28K510                                                        -->
<!--                                                                         -->
<!--  Change history:                                                        -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                           -->
<!--                                                                         -->
```

```
<!--==========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputlistserver [
<!ELEMENT inputlistserver EMPTY>
<!ATTLIST inputlistserver
 conversationname CDATA #REQUIRED
 lrmname CDATA #REQUIRED
 lrminame CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputlistserver
 conversationname = ''
 lrmname = ''
 lrminame = ''
/>
```

inputassociatelrmwithcontainer.xml

```
<?xml version='1.0'?>
<!--==========================================================================-->
<!--   File name:  inputassociatelrmwithcontainer.xml                         -->
<!--                                                                          -->
<!--   Descriptive name:  ...                                                 -->
<!--                                                                          -->
<!--   Proprietary statement:                                                 -->
<!--                                                                          -->
<!-- Licensed Material - Property of IBM                                      -->
<!--                                                                          -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                    -->
<!-- All Rights Reserved.                                                     -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or         -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.        -->
<!-- Status = H28K510                                                         -->
<!--                                                                          -->
<!--  Change history:                                                         -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                            -->
<!--                                                                          -->
<!--==========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputassociatelrmwithcontainer [
<!ELEMENT inputassociatelrmwithcontainer EMPTY>
<!ATTLIST inputassociatelrmwithcontainer
 conversationname CDATA #REQUIRED
 servername CDATA #REQUIRED
 lrmname CDATA #REQUIRED
 containername CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputassociatelrmwithcontainer
 conversationname = ''
 servername = ''
 lrmname = ''
 containername = ''
/>
```

inputdisassociatelrmfromcontainer.xml

```
<?xml version='1.0'?>
<!--=====================================================================-->
<!--   File name:  inputdisassociatelrmfromcontainer.xml                 -->
<!--                                                                     -->
<!--   Descriptive name:  ...                                            -->
<!--                                                                     -->
<!--   Proprietary statement:                                            -->
<!--                                                                     -->
<!-- Licensed Material - Property of IBM                                 -->
<!--                                                                     -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                               -->
<!-- All Rights Reserved.                                                -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or    -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.   -->
<!-- Status = H28K510                                                    -->
<!--                                                                     -->
<!--  Change history:                                                    -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                       -->
<!--                                                                     -->
<!--=====================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputdisassociatelrmfromcontainer [
<!ELEMENT inputdisassociatelrmfromcontainer EMPTY>
<!ATTLIST inputdisassociatelrmfromcontainer
 conversationname CDATA #REQUIRED
 servername CDATA #REQUIRED
 lrmname CDATA #REQUIRED
 containername CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputdisassociatelrmfromcontainer
 conversationname = ''
 servername = ''
 lrmname = ''
 containername = ''
/>
```

inputlistlrmassociatedwithcontainer.xml

```
<?xml version='1.0'?>
<!--=====================================================================-->
<!--   File name:  inputlistlrmassociatedwithcontainer.xml               -->
<!--                                                                     -->
<!--   Descriptive name:  ...                                            -->
```

```
<!--                                                                        -->
<!--   Proprietary statement:                                               -->
<!--                                                                        -->
<!-- Licensed Material - Property of IBM                                    -->
<!--                                                                        -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                  -->
<!-- All Rights Reserved.                                                   -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or       -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.      -->
<!-- Status = H28K510                                                       -->
<!--                                                                        -->
<!--  Change history:                                                       -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                          -->
<!--                                                                        -->
<!--==========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputlistlrmassociatedwithcontainer [
<!ELEMENT inputlistlrmassociatedwithcontainer EMPTY>
<!ATTLIST inputlistlrmassociatedwithcontainer
 conversationname CDATA #REQUIRED
 servername CDATA #REQUIRED
 lrmname CDATA #REQUIRED
 containername CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputlistlrmassociatedwithcontainer
 conversationname = ''
 servername = ''
 lrmname = ''
 containername = ''
/>
```

inputimportapplicationfamily.xml

```
<?xml version='1.0'?>
<!--==========================================================================-->
<!--   File name:  inputimportApplicationfamily.xml                         -->
<!--                                                                        -->
<!--   Descriptive name:  ...                                               -->
<!--                                                                        -->
<!--   Proprietary statement:                                               -->
<!--                                                                        -->
<!-- Licensed Material - Property of IBM                                    -->
<!--                                                                        -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                  -->
<!-- All Rights Reserved.                                                   -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or       -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.      -->
<!-- Status = H28K510                                                       -->
<!--                                                                        -->
<!--  Change history:                                                       -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                          -->
<!--                                                                        -->
```

```
<!--=========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputimpurtapplicationfamily [
<!ELEMENT inputimportApplicationFamily EMPTY>
<!ATTLIST inputimportApplicationFamily
  conversationname CDATA #REQUIRED
  sysplexname CDATA #REQUIRED
  servername CDATA #REQUIRED
  applicationfamilyname CDATA #REQUIRED
  ddlfilename CDATA #REQUIRED
  outputfilename CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputimportApplicationFamily
  conversationname = ''
  servername = ''
  applicationfamilyname = ''
  ddlfilename = ''
  outputfilename = ''
/>
```

inputremoveapplicationfamily.xml

```
<?xml version='1.0'?>
<!--=========================================================================-->
<!--   File name:  inputremoveApplicationfamily.xml                          -->
<!--                                                                         -->
<!--   Descriptive name:  ...                                                -->
<!--                                                                         -->
<!--   Proprietary statement:                                                -->
<!--                                                                         -->
<!-- Licensed Material - Property of IBM                                     -->
<!--                                                                         -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                                   -->
<!-- All Rights Reserved.                                                    -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or        -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.       -->
<!-- Status = H28K510                                                        -->
<!--                                                                         -->
<!--  Change history:                                                        -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                           -->
<!--                                                                         -->
<!--=========================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputremoveApplicationFamily [
<!ELEMENT inputremoveApplicationFamily EMPTY>
<!ATTLIST inputremoveApplicationFamily
  conversationname CDATA #REQUIRED
  servername CDATA #REQUIRED
  applicationfamilyname CDATA #REQUIRED
>
]>
```

```
<!--begin of default values-->
<inputremoveApplicationFamily
  conversationname = ''
  servername = ''
  applicationfamilyname = ''
/>
```

inputlistapplicationfamily.xml

```
<?xml version='1.0'?>
<!--=====================================================================-->
<!--   File name:  inputlistApplicationfamily.xml                       -->
<!--                                                                    -->
<!--   Descriptive name:  ...                                           -->
<!--                                                                    -->
<!--   Proprietary statement:                                           -->
<!--                                                                    -->
<!-- Licensed Material - Property of IBM                                -->
<!--                                                                    -->
<!-- 5655-A98 (C) Copyright IBM Corp. 2000                              -->
<!-- All Rights Reserved.                                               -->
<!-- U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or   -->
<!-- Disclosure restricted by GSA-ADP schedule contract with IBM Corp.  -->
<!-- Status = H28K510                                                   -->
<!--                                                                    -->
<!--  Change history:                                                   -->
<!--$L0=OW44455, H28K510, 20000721, PDCG: Created.                      -->
<!--                                                                    -->
<!--=====================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputlistApplicationFamily [
<!ELEMENT inputlistApplicationFamily EMPTY>
<!ATTLIST inputlistApplicationFamily
  conversationname CDATA #REQUIRED
  servername CDATA #REQUIRED
  applicationfamilyname CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputlistApplicationFamily
  conversationname = ''
  servername = ''
  applicationfamilyname = ''
/>
```

inputprocessearfile.xml

```
<?xml version='1.0'?>
<!--=====================================================================-->
<!--   File name:  inputprocessearfile.xml                              -->
<!--                                                                    -->
<!--   Descriptive name:  ...                                           -->
```

```
<!--                                                                      -->
<!--                                                                      -->
<!--   Proprietary statement:                                             -->
<!--                                                                      -->
<!--   Licensed Material - Property of IBM                                -->
<!--   5655-F31 (C) Copyright IBM Corp. 2000, 2001                        -->
<!--                                                                      -->
<!--   All Rights Reserved.                                               -->
<!--                                                                      -->
<!--   U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or   -->
<!--   Disclosure restricted by GSA-ADP schedule contract with IBM Corp.  -->
<!--                                                                      -->
<!--   Status = H28W400                                                   -->
<!--                                                                      -->
<!--                                                                      -->
<!--  Change history:                                                     -->
<!--$L0=OW44455, H28K510, 20010125, PDBL: Created.                        -->
<!--                                                                      -->
<!--======================================================================-->
<!-- internal DTD -->
<!DOCTYPE inputimpurtapplicationfamily [
<!ELEMENT inputprocessearfile EMPTY>
<!ATTLIST inputprocessearfile
  conversationname CDATA #REQUIRED
  j2eeservername CDATA #REQUIRED
  earfilename CDATA #REQUIRED
>
]>

<!--begin of default values-->
<inputprocessearfile
  conversationname = ''
  j2eeservername = ''
  earfilename = ''
/>
```

Chapter 10. Sample REXX scripts

Change attributes of active server

This example changes the attribute of an active server. Therefore a new conversation must be added, the server that should be modified must be listed to get the existing attributes, the changes must be performed and then the conversation has to activated.

```
/* REXX ----------------------------------------------------------- */
/* ================================================================ */
/*                                                                  */
/* COPYRIGHT =                                                      */
/* Licensed Material - Property of IBM                              */
/*                                                                  */
/* 5655-A98 (C) Copyright IBM Corp. 2000                            */
/* All Rights Reserved.                                             */
/* U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or */
/* Disclosure restricted by GSA-ADP schedule contract with IBM Corp.*/
/* Status = H28K510                                                 */
/*                                                                  */
/* FILENAME: SMAPI001                                               */
/*                                                                  */
/*                                                                  */
/* FUNCTION:                                                        */
/* Change attributes of active server with SM Scripting API         */
/*                                                                  */
/* !! WARNING !!                                                    */
/* This script changes the atributes and activates the conversation.*/
/* All changes will take place in the running system!               */
/*                                                                  */
/* ================================================================ */
/* This script changes attributes of an active server.              */
/* First a new conversation called "Demo Script 001" will be added. */
/* Then the server "BBOASR3" will be listed to get all properties.  */
/* After that the attributes values for "serverdescription" and     */
/* "garbagecollectioninterval" will be changed. Finally the         */
/* conversation "Demo Script 001" will be committed and activated   */
/*                                                                  */
/* Dependencies:                                                    */
/* The conversation "Demo Script 001" must not be added previously. */
/* The server "BBOASR3" must be valid in the current active         */
/* conversation.                                                    */

call syscalls 'ON'
signal on error

say "starting Demo Script 001"

/* 1. Step - Create new conversation"*/
say "Creating conversation..."
```

```
name. = 0
name.1 = "conversationname"

val. = 0
val.1 = "Demo Script 001"

rc = 0
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "Demo Script 001 createconversation failed while XMLGEN"
    exit
  end
  i = i+1
end;
rc = CB390CFG("-action 'createconversation' -xmlinput 'inputcreateconversation.xml'
               -input 'tempin' -output 'tempout'")
if (rc == 4) then do
  say "Demo Script 001 createconversation failed"
  exit
end
say "conversation created"
/* 1. Step - End"*/


/* 2. Step - Change active server"*/
say "Changing server..."

sval. = 0
sname. = 0

name. = 0
name.1 = "conversationname"
name.2 = "servername"

val. = 0
val.1 = "Demo Script 001"
val.2 = "BBOASR3"

rc = 0
i = 1
l = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "Demo Script 001 listserver failed while XMLGEN"
    exit
  end
  i = i+1
end;

rc = CB390CFG("-action 'listserver' -xmlinput 'inputlistserver.xml'
               -input 'tempin' -output 'tempout'")
if (rc == 4) then do
  say "Demo Script 001 listserver failed"
  exit
end

do forever
  n = XMLEXTRACT("tempout" l "N")
  if n <> '0' then do
    sname.l = n
    if n = "serverdescription" then do
      sval.l = "New Description"
    end
    else if n = "garbagecollectioninterval" then do
      sval.l = "55555"
    end
    else do
      v = XMLEXTRACT("tempout" l "V")
      sval.l = v
    end
    rc = XMLGEN("tempin" sname.l sval.l)
    if (rc == 4) then do
      say "Demo Script 001 changeserver failed while XMLGEN"
      exit
    end
  end
  else
    leave
  l=l+1
end

rc = CB390CFG("-action 'changeserver' -xmlinput 'inputchangeserver.xml'
               -input 'tempin' -output 'tempout'")
if (rc == 4) then do
  say "Demo Script 001 changeserver failed"
  exit
end
say "Server changed"
/* 2. Step - End"*/

/* 3. Step - Commit and activate conversation"*/
say "Committing conversation..."
name. = 0
name.1 = "conversationname"

val. = 0
val.1 = "Demo Script 001"

rc = 0
i = 1

do while(name.i <> '0')
  rc = XMLGEN("tempin" name.i val.i)
  if (rc == 4) then do
    say "Demo Script 001 commitconversation failed while XMLGEN"
    exit
  end
  i = i+1
end;
rc = CB390CFG("-action 'commitconversation' -xmlinput 'inputcommitconversation.xml'
               -input 'tempin' -output 'tempout'")
if (rc == 4) then do
  say "Demo Script 001 commitconversation failed"
  exit
end
say "Conversation committed and activated"
/* 3. Step - End"*/

say "Demo Script 001 completed"
exit

error:
say "Error" rc "at line" sigl
say sourceline(sigl)
exit
```

Add container and LRM to active server

This example adds a container and a LRM to the active server. Therefore a new conversation must be added, the container and the LRM must be added and the conversation has to be activated.

```
/* REXX ----------------------------------------------------------- */
/* ================================================================ */
/*                                                                  */
/* COPYRIGHT =                                                      */
/* Licensed Material - Property of IBM                              */
/*                                                                  */
/* 5655-A98 (C) Copyright IBM Corp. 2000                            */
/* All Rights Reserved.                                             */
/* U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or */
/* Disclosure restricted by GSA-ADP schedule contract with IBM Corp.*/
/* Status = H28K510                                                 */
/*                                                                  */
/* FILENAME: SMAPI002                                               */
/*                                                                  */
/*                                                                  */
/* FUNCTION:                                                        */
/* Add a new container and a new LRM to the running system          */
/*                                                                  */
/* !! WARNING !!                                                    */
/* This script adds a new container and a new LRM to the active     */
/* conversation. All changes will take place in the running system. */
/*                                                                  */
/* ================================================================ */
/* This script adds a new container "Demo_Container" and a new LRM  */
/* "Demo_LRM" to the running system. First a new conversation       */
/* "Demo Script 002" will be added. Then the new container          */
/* "Demo_Container" will be added following the LRM "Demo_LRM"      */
/* will be added too. Finally the conversation "Demo Script 002"    */
/* will be committed and activated.                                */
/*                                                                 */
/* Dependencies:                                                   */
/* The conversation "Demo Script 002" must not be added previously. */
/* The server "BBOASR3" must be valid in the current active        */
/* conversation.                                                   */

call syscalls 'ON'
signal on error


say "starting Demo Script 002"

/* 1. Step - Create new conversation"*/
say "Creating new conversation..."
name. = 0
name.1 = "conversationname"

val. = 0
val.1 = "Demo Script 002"

rc = 0
i = 1

do while(name.i <> '0')
    rc = XMLGEN("tempin" name.i val.i)
    if (rc == 4) then do
      say "Demo Script 002 createconversation failed while XMLGEN"
      exit
    end
    i = i+1
end;
rc = CB390CFG("-action 'createconversation' -xmlinput 'inputcreateconversation.xml'
               -input 'tempin' -output 'tempout'")
if (rc == 4) then do
  say "Demo Script 002 createconversation failed"
  exit
end
say "Conversation created"
/* 1. Step - End"*/


/* 2. Step - Adding container"*/
say "Adding container..."
name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "containername"
name.4 = "containerdescription"
name.5 = "aclcheckrequired"
name.6 = "activationisolationpolicy"
name.7 = "passivationconstraints"
name.8 = "managedobjectrefreshpolicy"
name.9 = "transactionpolicy"
```

```
val. = 0
val.1 = "Demo Script 002"
val.2 = "BBOASR3"
val.3 = "Demo_Container"
val.4 = "Demo Container Description"
val.5 = "N"
val.6 = "Transaction_Level"
val.7 = "Not_Pinned"
val.8 = "At_Activation"
val.9 = "TX_Required"

rc = 0
i = 1

do while(name.i <> '0')
    rc = XMLGEN("tempin" name.i val.i)
    if (rc == 4) then do
      say "Demo Script 002 createcontainer failed while XMLGEN"
      exit
    end
    i = i+1
end;
rc = CB390CFG("-action 'createcontainer' -xmlinput 'inputcreatecontainer.xml'
               -input 'tempin' -output 'tempout'")
if (rc == 4) then do
  say "Demo Script 002 createcontainer failed"
  exit
end
say "Container added"
/* 2. Step - End"*/

/* 3. Step - Adding LRM"*/
say "Adding LRM..."
name. = 0
name.1 = "conversationname"
name.2 = "lrmname"
name.3 = "lrmdescription"
name.4 = "coclassname"
name.5 = "codllname"
name.6 = "coclasscreatefunction"
name.7 = "lrmsubsystemtype"

val. = 0
val.1 = "Demo Script 002"
val.2 = "Demo_LRM"
val.3 = "Demo LRM Description"
val.4 = ""
val.5 = ""
val.6 = ""
val.7 = "DB2"

rc = 0
i = 1

do while(name.i <> '0')
```

```
    rc = XMLGEN("tempin" name.i val.i)
    if (rc == 4) then do
      say "Demo Script 002 createlrm failed while XMLGEN"
      exit
    end
    i = i+1
end;
rc = CB390CFG("-action 'createlrm' -xmlinput 'inputcreatelrm.xml'
               -input 'tempin' -output 'tempout'")
if (rc == 4) then do
  say "Demo Script 002 createlrm failed"
  exit
end
say "LRM added"
/* 3. Step - End"*/


/* 4. Step - Commit and activate conversation"*/
name. = 0
name.1 = "conversationname"

val. = 0
val.1 = "Demo Script 002"

rc = 0
i = 1

do while(name.i <> '0')
    rc = XMLGEN("tempin" name.i val.i)
    if (rc == 4) then do
      say "Demo Script 002 commitconversation failed while XMLGEN"
      exit
    end
    i = i+1
end;
rc = CB390CFG("-action 'commitconversation' -xmlinput 'inputcommitconversation.xml'
               -input 'tempin' -output 'tempout'")
if (rc == 4) then do
  say "Demo Script 002 commitconversation failed"
  exit
end
/* 4. Step - End"*/

say "Demo Script 002 completed"
exit

error:
say "Error" rc "at line" sigl
say sourceline(sigl)
exit
```

Delete application from active server

This example deletes an application from the active server. Therefore a new conversation must be added, the application must be deleted and then the conversation has to activated.

```
/* REXX ------------------------------------------------------------ */
/* ================================================================= */
/*                                                                   */
/* COPYRIGHT =                                                       */
/* Licensed Material - Property of IBM                               */
/*                                                                   */
/* 5655-A98 (C) Copyright IBM Corp. 2000                             */
/* All Rights Reserved.                                              */
/* U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or */
/* Disclosure restricted by GSA-ADP schedule contract with IBM Corp.*/
/* Status = H28K510                                                  */
/*                                                                   */
/* FILENAME: SMAPI003                                                */
/*                                                                   */
/*                                                                   */
/* FUNCTION:                                                         */
/* Delete application family from active server                     */
/*                                                                   */
/* !! WARNING !!                                                     */
/* This script deletes the application family and activates the     */
/* conversation. All changes will take place in the running system. */
/*                                                                   */
/* ================================================================= */
/* This script deletes an application family from the active        */
/* server. First a new conversation "Demo Script 003" will be added.*/
/* Then the application family "WAREHOUSES3" will be deleted and    */
/* finally the conversation "Demo Script 003" will be committed and */
/* activated                                                         */
/*                                                                   */
/* Dependencies:                                                     */
/* The conversation "Demo Script 003" must not be added previously. */
/* The server "BBOASR3" must be valid in the current active         */
/* conversation and the application family "WAREHOUSES3" must be    */
/* present in the server "BBOASR3"                                   */

call syscalls 'ON'
signal on error


say "starting Demo Script 003"

/* 1. Step - Create new conversation"*/
say "Creating new conversation..."
name. = 0
name.1 = "conversationname"

val. = 0
val.1 = "Demo Script 003"
```

```
rc = 0
i = 1

do while(name.i <> '0')
    rc = XMLGEN("tempin" name.i val.i)
    if (rc == 4) then do
      say "Demo Script 003 createconversation failed while XMLGEN"
      exit
    end
    i = i+1
end;
rc = CB390CFG("-action 'createconversation' -xmlinput 'inputcreateconversation.xml'
               -input 'tempin' -output 'tempout'")
if (rc == 4) then do
  say "Demo Script 003 createconversation failed"
  exit
end
say "Conversation created"
/* 1. Step - End"*/


/* 2. Step - Deleting application"*/
say "Deleting application..."
name. = 0
name.1 = "conversationname"
name.2 = "servername"
name.3 = "applicationfamilyname"

val. = 0
val.1 = "Demo Script 003"
val.2 = "BBOASR3"
val.3 = "WAREHOUSES3"

rc = 0
i = 1

do while(name.i <> '0')
    rc = XMLGEN("tempin" name.i val.i)
    if (rc == 4) then do
      say "Demo Script 003 removeApplicationfamily failed while XMLGEN"
      exit
    end
    i = i+1
end;
rc = CB390CFG("-action 'removeApplicationfamily' -xmlinput 'inputremoveApplicationfamil
               -input 'tempin' -output 'tempout'")
if (rc == 4) then do
  say "Demo Script 003 removeApplicationfamily failed"
  exit
end
say "Application deleted"
/* 2. Step - End"*/


/* 3. Step - Commit and activate conversation"*/
```

```
name. = 0
name.1 = "conversationname"

val. = 0
val.1 = "Demo Script 003"

rc = 0
i = 1

do while(name.i <> '0')
    rc = XMLGEN("tempin" name.i val.i)
    if (rc == 4) then do
      say "Demo Script 003 commitconversation failed while XMLGEN"
      exit
    end
    i = i+1
end;
rc = CB390CFG("-action 'commitconversation' -xmlinput 'inputcommitconversation.xml'
               -input 'tempin' -output 'tempout'")
if (rc == 4) then do
  say "Demo Script 003 commitconversation failed"
  exit
end
/* 3. Step - End"*/

say "Demo Script 003 completed"
exit

error:
say "Error" rc "at line" sigl
say sourceline(sigl)
exit
```

Appendix. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will

Examples in this book

Disclaimer - Programming Interface information

This publication documents information that is NOT intended to be used as Programming Interfaces of WebSphere for z/OS.

Trademarks

Glossary

For more information on terms used in this book, refer to one of the following sources:

- *WebSphere Application Server V4.0 for z/OS and OS/390 Glossary*, SC09-4450, located on the Internet at:

  `http://www.ibm.com/software/webservers/appserv/`
- Sun Microsystems Glossary of Java Technology-Related Terms, located on the Internet at:

  `http://java.sun.com/docs/glossary.html`

If you do not find the term you are looking for, refer to *IBM Glossary of Computing Terms*, located on the Internet at:

`http://www.ibm.com/ibm/terminology/`

or the Sun Web site, located on the Internet at:

`http://www.sun.com/`

Program Number: 5655–F31

Printed in the United States of America

SA22-7839-01