



iSeries

# System Manager for iSeries Commands

*Version 5 Release 3*







@server

iSeries

System Manager for iSeries Commands

*Version 5 Release 3*

**Note**

Before using this information and the product it supports, be sure to read the information in "Notices," on page 413.

**First Edition (May 2004)**

This edition applies to version 5, release 3, modification 0 of System Manager for iSeries (product number 5722-SM1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CICS models.

© Copyright International Business Machines Corporation 1998, 2004. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

## Contents

Add Command CRQ Activity (ADDCMDCRQA) . . . . .	1	Copy PTF to Save File (CPYPTFSAVF)	225
Add CRQ Activity (ADDCRQA) . . . . .	13	Create Product Definition (CRTPRDDFN) . . . . .	229
Add License CRQ Activity (ADDLICCRQA) . . . . .	41	Create Product Load (CRTPRDLOD)	235
Add Object CRQ Activity (ADDOBJCRQA). . . . .	49	Create Program Temporary Fix (CRTPTF) . . . . .	245
Add Product CRQ Activity (ADDPRDCRQA) . . . . .	67	Create PTF Package (CRTPTFPKG)	255
Add License Information (ADDPRDLICI)	79	Delete Product Definition (DLTPRDDFN) . . . . .	261
Add PTF CRQ Activity (ADDPTFCRQA)	85	Delete Product Load (DLTPRDLOD)	263
Add Resource CRQ Activity (ADDRSCCRQA) . . . . .	97	Delete Remote PTF (DLTRMTPTF) . . . . .	267
Apply Remote PTF (APYRMTPTF) . . . . .	107	Delete Submitted CRQ (DLTSBMCRQ)	271
Change Command CRQ Activity (CHGCMDCRQA) . . . . .	115	Delete SMG Object (DLTSMGOBJ) . . . . .	273
Change CRQ Activity (CHGCRQA)	127	Display Submitted CRQs (DSPSBMCRQ). . . . .	277
Change License CRQ Activity (CHGLICCRQA) . . . . .	157	Display Submitted CRQ Act (DSPSBMCRQA) . . . . .	283
Change Object CRQ Activity (CHGOBJCRQA) . . . . .	165	Display Submitted CRQ Messages (DSPSBMCRQM) . . . . .	287
Change Product CRQ Activity (CHGPRDCRQA) . . . . .	183	Display Service Provider Attr (DSPSRVPVDA) . . . . .	293
Change Product Object Desc (CHGPRDOBJD) . . . . .	195	End Submitted CRQ Activity (ENDSBMCRQA) . . . . .	295
Change PTF CRQ Activity (CHGPTFCRQA) . . . . .	199	End System Manager (ENDSYSMGR)	299
Change Resource CRQ Activity (CHGRSCCRQA) . . . . .	211	Generate License Key (GENLICKEY)	301
Change Service Provider Attr (CHGSRVPVDA) . . . . .	221	Hold Program Temporary Fix (HLDPTF) . . . . .	307
		Hold Submitted CRQ Activity (HLDSBMCRQA) . . . . .	309
		Install Remote Product (INSRMTPRD)	311

<b>Order Supported Product PTFs (ORDSPTPTF)</b> . . . . .	<b>317</b>	<b>Send License (SNDLIC)</b> . . . . .	<b>367</b>
<b>Package Product Option (PKGPRDOPT)</b> . . . . .	<b>319</b>	<b>Send Product (SNDPRD)</b> . . . . .	<b>371</b>
<b>Release Program Temporary Fix (RLSPTF)</b> . . . . .	<b>323</b>	<b>Send Program Temporary Fix (SNDPTF)</b> . . . . .	<b>377</b>
<b>Release Submitted CRQ Activity (RLSSBMCRQA)</b> . . . . .	<b>325</b>	<b>Send SMG Object (SNDSMGOBJ)</b> . . .	<b>387</b>
<b>Remove CRQD Activity (RMVCRQDA)</b>	<b>329</b>	<b>Start System Manager (STRSYSMGR)</b>	<b>393</b>
<b>Remove Remote PTF (RMVRMTPTF)</b>	<b>331</b>	<b>Work with CRQ Descriptions (WRKCRQD)</b> . . . . .	<b>395</b>
<b>Restore APAR Data (RSTAPARDTA)</b>	<b>339</b>	<b>Work with PTF (WRKPTF)</b> . . . . .	<b>397</b>
<b>Retrieve Product (RTVPRD)</b> . . . . .	<b>341</b>	<b>Work with Submitted CRQs (WRKSBMCRQ)</b> . . . . .	<b>399</b>
<b>Retrieve Program Temporary Fix (RTVPTF)</b> . . . . .	<b>345</b>	<b>Work with Submitted CRQ Act (WRKSBMCRQA)</b> . . . . .	<b>405</b>
<b>Retrieve SMG Object (RTVSMGOBJ)</b>	<b>349</b>	<b>Work with Supported Products (WRKSPTPRD)</b> . . . . .	<b>409</b>
<b>Run SMG Command (RUNSMGCMD)</b>	<b>353</b>	<b>Work with Service Requesters (WRKSRVRQS)</b> . . . . .	<b>411</b>
<b>Run SMG Object (RUNSMGOBJ)</b> . . .	<b>359</b>	<b>Appendix. Notices</b> . . . . .	<b>413</b>
<b>Submit Change Request (SBMCRQ)</b>	<b>363</b>		

---

## Add Command CRQ Activity (ADDCMDCRQA)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Add Command Change Request Activity (ADDCMDCRQA) command adds an activity to a change request description to run a command on one or more managed systems.

The activity can be conditioned so that it only runs after one or more other activities have completed (successfully or unsuccessfully). The activity can also be scheduled to run at a date and time in the future.

### Restrictions:

1. You must have \*CHANGE authority to the change request description and \*EXECUTE authority to the library.
2. If a node list (NODL) value is specified, the node list can only contain entries that have a value of \*SNA for the address type.

### Notes:

The following notes provide information on how the command works.

1. Authorization to the product specified on the activity is not verified until the activity runs.
2. All conditions must be satisfied before the activity can run.
3. The start times indicate when the activity can be started. Actual start times can be later due to network and system delays.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACTIVITY	Activity	<i>Name, *GEN, *LAST</i>	Optional, Positional 2
CMD	Command to run	<i>Command string</i>	Optional, Positional 3
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: <b>*NONE</b> Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Keyword	Description	Choices	Notes
CPNAME	Managed system node names	Single values: *LOCAL, *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name</i> , *NETATR	
	Element 2: Control point	<i>Communications name</i>	
RMTSTRTIME	Start time on managed system	<i>Element list</i>	Optional
	Element 1: Time zone	*LCLSYS, *MGDSYS	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *CURRENT	
	Element 2: Date	<i>Date</i> , *CURRENT, *NEXT	
	Element 3: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *ANY, *CURRENT	
	Element 2: Date	<i>Date</i> , *ANY, *CURRENT, *NEXT	
RTNSPLF	Return spooled file	*YES, *NO, *FAIL	Optional
USRPRF	User profile	<i>Simple name</i> , *NONE	Optional
PASSWORD	Password	<i>Character value</i> , *USRPRF, *NONE	Optional
ENCODE	Encode command	*YES, *NO	Optional
TEXT	Text 'description'	<i>Character value</i> , *GEN	Optional
COND	Activity conditions	Single values: *NONE Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Activity	<i>Generic name, name</i> , *PRV	
	Element 2: Relational operator	*EQ, *GT, *LT, *LE, *NE, *GE	
	Element 3: End code	0-99, *SUCCESS, *FAIL, *NOTRUN, *ANY	
	Element 4: Condition mode	*ALLNODES, *SAMENODE	
STRTIME	Activity start time	<i>Element list</i>	Optional
	Element 1: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *CURRENT	
	Element 2: Date	<i>Date</i> , *CURRENT, *NEXT	
	Element 2: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *ANY, *CURRENT	
	Element 2: Date	<i>Date</i> , *ANY, *CURRENT, *NEXT	
HOLD	Hold activity	*NO, *YES	Optional

Top

## Change request description (CRQD)

Specifies the change request description object name.

The possible library values are:

**\*LIBL** All of the libraries in your library list and in the system portions of the job's library list are searched.

**\*CURLIB**

The current library for the job is used to locate the object.



### *library-name*

Specify that only the library named in this parameter is searched.

The possible change request description value is:

### *change-request-description*

Specify the name of the change request description object.

Top

---

## Activity (ACTIVITY)

Specifies the name of the activity to add to the change request description.

**\*GEN** An activity name is generated. The activity name is of the form QACTxxxxxx where xxxxxx is the first multiple of ten not already being used.

### **\*LAST**

The activity is the last to run in the change request. When \*LAST is specified for the activity (ACTIVITY) parameter, the condition (COND) parameter and the start time (STRTIME) parameter cannot be specified. Only one activity named \*LAST can exist in the change request description.

### *activity-name*

Specify a 10-character activity name.

Top

---

## Command to run (CMD)

Specifies the CL command to run. The command can be any command that can be run in batch. The command must follow the OS/400 CL command format of 1 to 10 characters with the first character in alphabetic order and the other nine characters alphanumeric.

### *command-string*

Specify the command to run on the managed system. Command prompting support is available if you press F4.

Top

---

## Managed systems node list (NODL)

Specifies that the node list parameter is the object name that contains a list of systems which are the destinations for the activity. This parameter cannot be specified if the control point name (CPNAME) parameter is specified.

The possible values are:

### **\*NONE**

The systems on which this activity is to be performed are not specified by a node list. Individual control point names must be specified.

The possible values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

### **\*CURLIB**

The current library for the job is used to locate the node list object.

*library-name.*

Specify the name of the library to be searched.

The possible node list name value is:

*node-list-name*

Specify the node list object name containing the list of systems on which the activity is to be performed.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if the node list (NODL) parameter is specified.

The possible values are:

**\*LOCAL**

The local system is identified as the target system. If \*LOCAL is specified, the command is run on the local system. Any spooled files created remain on the system.

**\*NONE**

The systems on which this activity is to be performed are not identified individually. A node list must be specified.

**\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

*network-identifier*

Specify the APPN network identifier of the managed system on which the activity is to be performed.

The possible control point value is:

*control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed.

Top

---

## Start time on managed system (RMTSTARTIME)

Specifies the date and time when the activity can begin running on the managed system. The current date and time values, and the next date values are determined when the activity begins running at the central site system based on the central site date and time.

The possible time zone values are:

**\*LCLSYS**

The remote start time is specified in the time zone of the central site system.

**\*MGDSYS**

The remote start time is specified in the time zone of the managed system.

The possible start after time values are:

This is the definition of the time after which the activity is to start.

**\*CURRENT**

This function can start on the managed system at any time on or after the time this activity is started on the date specified in element 3.

***start-after-time***

Specify the time after when this function can start on the managed system. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where, hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

**\*CURRENT**

This function starts on the managed system on any date on or after the activity starts on the central site system.

**\*NEXT**

This function starts on the managed system on any date after the activity starts on the central site system.

***start-after-date***

Specify the date after the functions start on the managed system. The date must be specified in the job date format.

The possible start before time values are:

This element is ignored if start before time is \*ANY.

**\*ANY** The activity can start at any time on or after the start time.

**\*CURRENT**

The activity must start before the time at which the change request was submitted on the date specified on the start before date element.

***start-before-time***

Specify the time before which the activity must be started. If the activity cannot be started before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start before date values are:

**\*ANY** The activity can start at any date on or after the start date.

**\*CURRENT**

The activity must start on the date the change request was submitted.

**\*NEXT**

The activity must start by the day after the date the change request was submitted.

***start-before-date***

The date before which the activity must be started. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

**Note:**

The special values \*CURRENT and \*NEXT cannot be specified for the date and the time if the time zone value \*MGDSYS is specified.

Top

---

## Return spooled file (RTNSPLF)

Specifies whether the output spooled file from the remote command is returned.

The possible values are:

- \*YES** The spooled files created from the remote command are returned from the managed system. The spooled files from all of the nodes are combined into one spooled file that can be viewed by displaying the command activity details of the change request.
- \*NO** The output data is not returned from the managed system.
- \*FAIL** The spooled file job log is returned from the managed system if the command fails when running.

Top

---

## User profile (USRPRF)

Specifies the user profile under which the command runs at the remote systems. If the remote system is running the NetView Remote Operations Agent/400 product this parameter is ignored at the remote systems.

The possible values are:

- \*NONE**  
No user profile is specified. The default user profile is used on each managed system.

*user-profile*

Specify the name of the user profile.

Top

---

## Password (PASSWORD)

Specifies the password for the remote user profile.

- Managed Systems at releases prior to V5R1M0 only accept uppercased passwords up to 10 characters long. If a longer password is entered, SMU18A2 message with 100B0007 SNA sense code is returned, indicating that the request was rejected.
- Managed Systems at release V5R1M0 and above, running with QPWDLVL system value:
  - 0 or 1 — truncate the received passwords to 10 characters.
  - 2 or 3 — accept passwords up to 128 characters.

The possible values are:

- \*NONE**  
No password is specified.

- \*USRPRF**  
The password is the same as the user profile.

*password*

Specify the password for the user profile.

Top

---

## Encode command (ENCODE)

Specifies whether or not the command, user profile, and password are encoded when sent to the managed systems. If the managed system is running the NetView Remote Operations Agent/400 product, this parameter must be to \*NO.

The possible values are:

- \*NO** The command, user profile, and password are not encoded when the request is sent to the managed systems.
- \*YES** The command, user profile, and password are encoded when the request is sent to the managed systems. The remote command key, which is a managed system attribute, must be specified on both the central site system and the managed system. This attribute can be changed using the Change Managed System Attributes (CHGMGDSYSA) command. The remote command key must be the same on the central site system and the managed system.

Top

---

## Text 'description' (TEXT)

Specifies the activity description.

**\*GEN:** A description is generated based on the action specified.

*text-description*

Specify a 50-character description of the activity.

Top

---

## Activity conditions (COND)

Specifies which conditions must be met before this activity can be performed. Each condition identifies an activity which must run before this activity and the value the end code from that activity must have to allow this activity to run. The default condition is that the previous activity (in alphabetical order) must complete successfully before this activity can be run.

The possible conditioning activity values are:

The activity which must be run before this activity.

**\*PRV** This activity is conditioned on the previous activity. Activities are ordered alphabetically by activity name. If the activity being added is the first activity, a previous activity does not exist and any condition with \*PRV is marked as having been met.

*conditioning-activity-name*

Specify the name of this activity which must run before this activity. The activity name specified in the activity (ACTIVITY) parameter cannot be specified in the conditioning activity name. An activity cannot be conditioned on itself.

*generic\*-conditioning-activity-name*

Specify the generic name of the activities which must run before this activity.

The possible relational operator value is:

This element is the relational operator to use when comparing the end code from the conditioning activity.

**\*EQ** Equal

- \*GT** Greater than
- \*LT** Less than
- \*NE** Not equal
- \*GE** Greater than or equal
- \*LE** Less than or equal

The possible condition code value is:

This element is the value compared to the actual end code of the conditioning activity.

**\*SUCCESS**

The activity ended successfully (0 <= end code <= 9). This end code can only be specified with relational operator \*EQ or \*NE.

**\*FAIL** The activity failed (10 <= end code <= 89). This end code can only be specified with relational operator \*EQ or \*NE.

**\*NOTRUN**

The activity never started (90 <= end code <= 99). This end code is only specified with relational operator \*EQ or \*NE.

**\*ANY** The activity ended with any end code. This end code is only specified with relational operator \*EQ.

***end-code***

Specify an integer value (0-99) that indicates the result of an activity (success or failure). The end code ranges and descriptions are the following:

- 00** Activity completed successfully.
- 01-09** Activity completed with warning messages.
- 10-29** Activity did not complete successfully.
- 30-39** Activity was canceled by a user before it completed.
  - 30 = Activity ended with \*CNTRLD option
  - 35 = Activity ended with \*IMMED option
  - 39 = Activity ended with \*FRCFAIL option
- 40-49** Activity was not run due to errors detected by the application.
  - 40 = Activity not run for security reasons
- 90-99** Activity was not run because conditions or schedules were not met.
  - 95 = Scheduled start time expired
  - 99 = Conditions cannot be met

The possible condition mode values are:

This element indicates which systems the conditioning activity must have completed on before this activity can be performed.

**\*ALLNODES**

The conditioning activity specified must complete on all nodes before this activity can run.

**\*SAMENODE**

When the conditioning activity specified completes for a given node, the activity specified on the ACTIVITY parameter may run for that same node even though the conditioning activity specified

may not have completed on all other nodes. In the case where this activity lists a node not in the conditioning activity, this activity may run for that node; the condition is ignored.

The possible value is:

**\*NONE**

There are no conditions for this activity.

Top

---

## Activity start time (STRTIME)

Specifies the date and time when this activity can be started on the central site system. The current date and time values and next date values are determined when the change request is submitted.

The possible start after time values are:

**\*CURRENT**

This activity can start any time on or after the time when the change request is submitted.

***start-after-time***

Specify the time when this activity can start. The time can be entered as 4 or 6 digits (hhmm or hhmmss), where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

**\*CURRENT**

This activity can start on or after the date on which the change request is submitted.

**\*NEXT**

The activity can start on any date after the date the change request is submitted.

***start-after-date***

Specify the date after this activity can start. The date must be specified in the job date format.

The possible start before time values are:

This element is ignored if the start before date is \*ANY.

**\*ANY** The activity can start at any time on or before the start before date.

**\*CURRENT**

The activity must start before the time at which the change request was submitted on the date specified on the start before date element.

***start-before-time***

Specify the time before which the activity must start. If the activity cannot be started before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start before date values are:

**\*ANY** The activity can start at any time after the start after time and the start after date.

**\*CURRENT**

The activity must start on the date the change request is submitted.

### **\*NEXT**

The activity must start by the day after the date the change request is submitted.

### ***start-before-date***

Specify the date before the activity must start. If the activity is not started by this date, it never starts. The date must be specified in the job date format.

Top

---

## **Hold activity (HOLD)**

Specifies that the activity be held when the change request is submitted.

**\*NO** The activity is not held. It runs when all conditions are met at the start time.

**\*YES** The activity is held when the change request is submitted. It must be released by you before it runs.

Top

---

## **Examples**

### **Example 1: Adding an Activity**

```
ADDCMDCRQA CRQD(MYLIB/CR1) ACTIVITY(ACT01) CMD(STRSBS QCMN)
```

This example shows how activity ACT01 is added to change request description CR1 in library MYLIB to start the QCMN subsystem on the central site system. The activity runs as soon as the change request is submitted.

### **Example 2: Adding an Activity for a Node list**

```
ADDCMDCRQA CRQD(MYLIB/CR2) ACTIVITY(ACT01)
            CMD(STRSBS QSNADS) NODL(MYLIB/STORES)
            CPNAME(*NONE) RTNSPLF(*FAIL)
```

This example shows how activity ACT01 starts the subsystem QSNADS on the systems identified in the STORES node list in the library MYLIB. The example also asks for the spooled file to be returned to the central site system if the command fails.

### **Example 3: Adding an Activity for Two Systems**

```
ADDCMDCRQA CRQD(MYLIB/CR3) ACTIVITY(ACT01)
            CMD(PRODLIB/RUNREPORTS)
            CPNAME((STORENET STOREA) (STORENET STOREB))
            USRPRF(REPORTOPER) PASSWORD(OPERPASS)
            ENCODE(*YES)
```

This example shows how activity ACT01 runs the detail reports for STOREA and STOREB in the STORENET network. The report runs under the REPORTOPER user profile on the managed systems. Because the user profile and a password are supplied, the request will be encoded when sent to the managed system.

Top

---

## **Error messages**

None







---

## Add CRQ Activity (ADDCRQA)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Add Change Request Activity (ADDCRQA) command adds a change management activity to a change request description. The change management action that is to be performed by the added activity depends on the value specified on the action parameter.

The activity can be conditioned so that it will only run after one or more other activities have completed (successfully or unsuccessfully). The activity may also be scheduled to run at a date and time in the future.

### Restrictions:

- You must have \*CHANGE authority to the change request description object and \*EXECUTE authority to the library.
- If a node list (NODL) value is specified, the node list can only contain entries that have a value of \*SNA for the address type.
- Global names or component names must be 64 characters or less including one separator character between tokens.

**Note:** If you need to cancel an installation of a NetView Distribution Manager (NetView DM) change file or other non-OS/400 object, you must use the Add Change Management Activity (QNSADDCM) API.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACTIVITY	Activity	<i>Name, *GEN, *LAST</i>	Optional, Positional 2
ACTION	Action	<i>*ACP, *ACT, *DLT, *INS, *RMV, *RTV, *RUN, *SND, *SNDINS, *SNDRUN, *UNINS</i>	Optional, Positional 3

Keyword	Description	Choices	Notes
GLBNAME	Global name	<i>Element list</i>	Optional
	Element 1: Global name token 1	<i>Character value, *NETID, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 2: Global name token 2	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 3: Global name token 3	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 4: Global name token 4	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 5: Global name token 5	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 6: Global name token 6	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 7: Global name token 7	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 8: Global name token 8	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 9: Global name token 9	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 10: Global name token 10	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
COMPNAME	Component name	<i>Element list</i>	Optional
	Element 1: Component name token 1	<i>Character value, *NETID</i>	
	Element 2: Component name token 2	<i>Character value, *NETID, *CPNAME</i>	
	Element 3: Component name token 3	<i>Character value, *NETID, *CPNAME</i>	
	Element 4: Component name token 4	<i>Character value, *NETID, *CPNAME</i>	
	Element 5: Component name token 5	<i>Character value, *NETID, *CPNAME</i>	
	Element 6: Component name token 6	<i>Character value, *NETID, *CPNAME</i>	
	Element 7: Component name token 7	<i>Character value, *NETID, *CPNAME</i>	
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: <b>*NONE</b> Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CPNAME	Managed system node names	Single values: <b>*NONE</b> Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Character value</i>	
TEXT	Text 'description'	<i>Character value, *GEN, *BLANK</i>	Optional
ACTFRC	Activation force	Single values: <b>*YES</b> Other values: <i>Element list</i>	Optional
	Element 1: Activation force	<b>*NO</b>	
	Element 2: Delay units	<b>*SECONDS, *MINUTES, *HOURS</b>	
	Element 3: Delay period	1-65535, <b>3600</b>	

Keyword	Description	Choices	Notes
ACTUSEACT	Activation use on activate	*BOTH, *PROD, *LAST, <u>*NONE</u>	Optional
CPRTYPE	Compression type	*ADAPTIVE, *SNA, *USER, <u>*NONE</u>	Optional
SNACPRCHR	SNA compression character	Hexadecimal value, <u>*BLANK</u>	Optional
USRCPRINF	User compression information	Element list	Optional
	Element 1: User compression name	Name	
	Element 2: User parameters	Character value	
CPRSTGSTT	Compression storage state	*DECOMPRESS, *COMPRESS, <u>*NONE</u>	Optional
CPRTFRSTT	Compression transfer state	*DECOMPRESS, *COMPRESS, <u>*NONE</u>	Optional
ACTUSEINS	Activation use on install	<u>*PROD</u> , *TRIAL	Optional
ALTACTCOMP	Alter active components	*ALLOWED, *NOTALLOWED, <u>*NONE</u>	Optional
AUTOACP	Auto accept	Single values: <u>*NONE</u> , *NO Other values: <i>Element list</i>	Optional
	Element 1: Auto accept	*YES, *DESIRED	
	Element 2: Delay days	0-255	
AUTORMV	Auto remove	*YES, *NO, *DESIRED, <u>*NONE</u>	Optional
PRETEST	Pre test	*YES, *NO, <u>*DESIRED</u>	Optional
POSTTEST	Post test	*YES, *NO, <u>*DESIRED</u>	Optional
ALWRMV	Allow remove	*NO, *DESIRED, <u>*YES</u>	Optional
PARM	Initiate parameters	Values (up to 40 repetitions): <i>Not restricted</i>	Optional
KEEP OBJ	Keep object	*YES, *NO, <u>*NONE</u>	Optional
COREQCHGNL	Corequisite change name list	Single values: <u>*NONE</u> Other values: <i>Element list</i>	Optional
	Element 1: Global name	Values (up to 6 repetitions): <i>Element list</i>	
	Element 1: Global name token 1	Character value	
	Element 2: Global name token 2	Character value	
	Element 3: Global name token 3	Character value	
	Element 4: Global name token 4	Character value	
	Element 5: Global name token 5	Character value	
	Element 6: Global name token 6	Character value	
	Element 7: Global name token 7	Character value	
	Element 8: Global name token 8	Character value	
	Element 9: Global name token 9	Character value	
	Element 10: Global name token 10	Character value	
REPLACE	Replace object	*YES, *ALLOWED, <u>*NO</u>	Optional

Keyword	Description	Choices	Notes
TODLTNAME	To be deleted name	Single values: <b>*NONE</b> Other values: <i>Element list</i>	Optional
	Element 1: To be deleted name token 1	<i>Character value</i> , *NETID, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 2: To be deleted name token 2	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 3: To be deleted name token 3	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 4: To be deleted name token 4	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 5: To be deleted name token 5	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 6: To be deleted name token 6	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 7: To be deleted name token 7	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 8: To be deleted name token 8	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 9: To be deleted name token 9	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 10: To be deleted name token 10	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
REFLVL	Refresh level	<i>Character value</i>	Optional
FRCUNINS	Force uninstall	*YES, <b>*NO</b>	Optional
RMTSTRTIME	Start time on managed system	Single values: <b>*NONE</b> Other values: <i>Element list</i>	Optional
	Element 1: Time zone	*LCLSYS, *MGDSYS	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , <b>*CURRENT</b>	
	Element 2: Date	<i>Date</i> , <b>*CURRENT</b> , *NEXT	
COND	Activity conditions	Single values: <b>*NONE</b> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Activity	<i>Generic name, name</i> , <b>*PRV</b>	
	Element 2: Relational operator	<b>*EQ</b> , *GT, *LT, *LE, *NE, *GE	
	Element 3: End code	0-99, <b>*SUCCESS</b> , *FAIL, *NOTRUN, *ANY	
	Element 4: Condition mode	<b>*ALLNODES</b> , *SAMENODE	
STRTIME	Activity start time	<i>Element list</i>	Optional
	Element 1: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , <b>*CURRENT</b>	
	Element 2: Date	<i>Date</i> , <b>*CURRENT</b> , *NEXT	
	Element 2: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , <b>*ANY</b> , *CURRENT	
	Element 2: Date	<i>Date</i> , <b>*ANY</b> , *CURRENT, *NEXT	
HOLD	Hold activity	<b>*NO</b> , *YES	Optional

Top

---

## Change request description (CRQD)

Specifies the change request description object name.

The possible values are:

**\*LIBL** All of the libraries in your library list and in the system portions of the job's library list are searched.

**\*CURLIB**  
The current library for the job is used to locate the object.

*library-name*  
Specify that only the library named in this parameter is searched.

The possible change request description value is:

*change-request-description*  
Specify the name of the change request description object.

Top

---

## Activity (ACTIVITY)

Specifies the name of the activity to add to the change request description.

**\*GEN** An activity name will be generated. The activity name is of the form QACTxxxxxx where xxxxxx is the first multiple of ten not already being used.

**\*LAST**  
The activity is the last to run in the change request. When \*LAST is specified for the activity (ACTIVITY) parameter, the condition (COND) parameter and the start time (STRTIME) parameter cannot be specified. Only one activity named \*LAST can exist in the change request description.

*activity-name*  
Specify a 10-character activity name.

Top

---

## Action (ACTION)

Specifies the change management action to be performed by the added activity.

**\*ACP** Causes the managed system to accept a previously installed object. The activity relinquishes resources at a managed system required to maintain removability of a change. This cancels the removability of a change previously installed in a removable manner. The resources released are, typically, unaltered versions of components affected by the change.

**\*ACT** Causes the managed system to activate all previously installed changes. Each managed system implements the activation in its own way. For example, the activity performs an initial program load (IPL) of a managed iSeries system or activates a configuration on a PS/2 and causes the PS/2 to restart.

**\*DLT** Requests a delete action at one or more managed systems.

**\*INS** Requests the installation of up to seven objects. The objects are treated as corequisites, which means that either all installations succeed or all do not. The activity uses an object and its corequisites, if any, to alter all components necessary to effect the change. The managed system can perform such alteration in a removable manner if required, so that a subsequent request

(remove) can return all those components to their original condition prior to the alteration. Also automatic removal or automatic acceptance is possible.

- \*RMV** The activity, when run at the managed system, returns (that is, removes) all components previously altered in connection with a change to their condition prior to the installation of the change. This is possible only for changes previously installed in a removable manner.
- \*RTV** Retrieves an object identified by its global name, from a managed system or from another central site system, for storage at the central site system. To retrieve an object from more than one system, a global name with an \*ANY token is required, so that each retrieved object has a unique global name. Global names with unspecified tokens (\*ANY, \*HIGHEST, or \*LOWEST) are stored in the distribution repository when they are retrieved.
- \*RUN** Causes a program or procedure to be run at one or more managed systems.
- \*SND** Sends an object from the central site system to one or more managed systems or to another central site system.
- \*SNDINS**  
Sends an object from the central site system to one or more managed systems and installs the object at the managed system.
- \*SNDRUN**  
Sends an object from the central site system to one or more managed systems and runs the object at the managed system.
- \*UNINS**  
Removes (that is, uninstalls) a previously installed component from a managed system. The component may have originally been built from the installation of multiple objects, all of which are removed.

Top

---

## Global name (GLBNAME)

Specifies a global name, which is a series of tokens that uniquely identify an object in an SNA network. The global name represents the name that is used to locate the appropriate catalog entry on both the central site system and the managed systems. The catalog entry specifies the object that is to be used on that system. For example, if a retrieve action is specified, the global name is used to determine the object that is to be retrieved on the managed system. Also, the global name shows the location where it is to be stored on the central site system.

Special values in a token position indicate how to search for the object. By specifying \*ANY in a token position, the token is ignored when searching for the correct object.

If an object is sent, the global name must have been previously cataloged so that it is associated with a local object name or associated with an object in the distribution repository. Retrieved objects, for which no catalog entry exists, are placed in the distribution repository.

Special values other than \*SERVER, \*TARGET, \*MDDATE, and \*MDTIME in a token position are allowed only for the following actions: \*DLT, \*RTV, \*SND, \*SNDRUN, or \*SNDINS. The special values allowed for the first token are \*NETID, \*SERVER, \*TARGET, \*MDDATE, and \*MDTIME.

### Element 1: Token 1

The possible values are:



**\*NETID**

The first global name token value is a network ID generated by the command from the network attributes. The network ID is determined by the current value of the LCLNETID network attribute value.

***global-name-token-1***

Specify the first token of the global name. The first token is recommended to be the registered enterprise ID or network ID.

**Element 2-10: Token 2-10**

The possible values are:

**\*ANY** Any token value matches when searching for the object where the action is to be performed.

**\*HIGHEST**

The object with the highest token value has the action performed on it. The token must be ordered. This useful when a token in a global name is used to indicate a different version of the object and you need to manipulate the object with the highest version level.

**\*LOWEST**

The object with the lowest token value has the action performed on it. The token must be ordered. This is useful when a token in global name is used to indicate a different version of the object and you need to manipulate the object with the lowest version level.

**\*NETID**

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

**\*CPNAME**

The control point name of this system is used. The control point name is determined by the current value of the LCLCPNAME network attribute value.

**\*SERVER:**

This token is stored within the change request activity with the value &SERVER, and is replaced by the short name of the change control server when the object is distributed.

**\*TARGET:**

This token is stored within the change request activity with the value &TARGET, and is replaced by the short name of the target when the object is distributed.

**\*MDDATE:**

This token is stored within the change request activity with the value &DATE, and is replaced when distributed by the date the object was last modified.

**\*MDTIME:**

This token is stored within the change request activity with the value &TIME, and is replaced when distributed by the time the object was last modified.

***global-name-token-n***

Specify one of a series of 1 to 16 character tokens that uniquely identify the object on which the action is to be performed. Characters A through Z and 0 through 9 can be used. Other special values (@, #, and \$) can be used for tokens that represent network IDs and system names.

**Note:** GLBNAME is valid only when ACTION(\*ACT) is not specified.

Top

---

## Component name (COMPNAME)

Component name, which is the set of global name tokens previous to the REF, FIX, or UPD subtree indicator (STI) tokens. The component name is used to identify the installable object that must be uninstalled from the managed system. The maximum number of tokens allowed is 7. The component name is used to identify the installable object that must be uninstalled from the managed system.

**Note:** COMPNAME is only valid when ACTION(\*UNINS) is specified.

### Element 1: Token 1

The only special value allowed for the first token is \*NETID.

The possible values are:

#### \*NETID

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

#### *component-name-token-1*

One of a series of 1 to 16 character tokens that uniquely identifies the object on which the action is to be performed. Characters A through Z and 0 through 9 can be used. Other special values (@, #, and \$) can be used for tokens that represent network IDs and system names.

#### **Elements 2-7: Tokens 2-7**

#### **\*NETID**

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

#### **\*CPNAME**

The control point name of this system is used. The network ID is determined by the current value of the LCLCPNAME network attribute value.

#### *component-name-token-n*

One of a series of 1 to 16 character tokens that uniquely identifies the object on which the action is to be performed. Characters A through Z and 0 through 9 can be used. Other special values (@, #, and \$) can be used for tokens that represent network IDs and system names.

Top

---

## Managed systems node list (NODL)

Specifies that the node list parameter is the object name that contains a list of systems that are the destinations for the activity. This parameter cannot be specified if the control point name (CPNAME) parameter is also specified.

The possible values are:

#### \*NONE

The systems on which this activity is to be performed are not specified by a node list. Individual control point names must be specified.

The possible values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

#### **\*CURLIB**

The current library for the job is used to locate the node list object.

*library-name*

Specify the name of the library to be searched.

*node-list-name*

Specify the node list object name containing the list of systems on which the activity is to be performed.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if the node list (NODL) parameter is specified. CPNAME is required unless NODL is specified.

The possible values are:

**\*NONE**

The systems on which this activity is to be performed are not identified individually. A node list must be specified.

**\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

*network-identifier*

Specify the APPN network identifier of the managed system on which the activity is to be performed. For NetView Distribution Management Agents, the network identifier is the change control server name.

*control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed. For NetView Distribution Management Agents, the control point name is the change control client which supports numeric characters (0-9) in the first position of control point names that are valid in other platforms.

Top

---

## Text 'description' (TEXT)

Specifies the activity description.

**\*GEN** A text description should be generated for the activity being added. The text description should be generated based the verb specified on the ACTION parameter. The text descriptions generated are:

- Accept object
- Restart the system
- Delete object
- Run object
- Install object
- Remove object
- Retrieve object
- Send object
- Send and run object

- Send and install object
- Uninstall object

**\*BLANK**

No text is specified.

*text-description*

Specify a 50-character textual description of the activity.

Top

## Activation force (ACTFRC)

Specifies whether or not the managed system should proceed with the activation based on its quiesced state.

*activation force*

Specify the activation force value.

\*NO The managed system will not proceed with the activation if the quiesce check shows that the managed system is still active.

*delay units*

Indicates the unit of time on which the delay period is specified.

\*SECONDS

The delay period will be specified in seconds.

\*MINUTES

The delay period will be specified in minutes.

\*HOURS

The delay period will be specified in hours.

*delay period*

Specifies the maximum amount of time that the managed system may wait to quiesce (if not already quiesced) before taking the action specified:

3600 The delay period default is 3600 seconds.

1-65535

Delay period range.

**\*YES** The managed system will proceed with the activation even if the quiesce check shows that the managed system is still active. Delay units and delay period are ignored when activation force is \*YES.

**Note:** ACTFRC is valid only when ACTION(\*ACT) is specified.

Top

## Activation use on activate (ACTUSEACT)

Activation use on activate. Specifies which components altered by changes will be used during the activation.

**\*NONE**

No activation use on activate is specified.

**\*BOTH**

Both trial and production version.

**\*PROD**

Production version only.

**\*LAST**

Last used; either both trial and production or production only.

**Note:** ACTUSEACT is valid only when ACTION(\*ACT) is specified.

Top

---

## Compression type (CPRTYPE)

Specifies the compression algorithm and related information associated with the compression of a particular change object. When one or both of the compression transfer state or compression storage state parameter specify \*COMPRESS, this parameter must be present and one (and only one) of the adaptive compression, SNA compression, or user compression algorithms may be requested. This support is dependent on the implementation on the managed system.

**\*NONE**

No compression type is specified.

**\*ADAPTIVE**

Specifies whether or not adaptive compression pertains to the requested object. The iSeries system does not support \*ADAPTIVE.

**\*SNA** Specifies whether or not SNA compression pertains to the requested object.

**\*USER**

Specifies whether or not a named user compression pertains to the requested object.

**Notes:**

1. CPRTYPE is valid only when ACTION(\*RTV), ACTION(\*SND), ACTION(\*SNDRUN), or ACTION(\*SNDINS) is specified
2. CPRTYPE cannot be specified when CMPSTGSTT and CPRTFRSTT are not specified.
3. CPRTYPE cannot be specified when CMPSTGSTT is \*DECOMPRESS and CPRTFRSTT is not specified.
4. CPRTYPE cannot be specified when CPRSTGSTT is not specified and CPRTFRSTT is \*DECOMPRESS.
5. CPRTYPE cannot be specified when CPRSTGSTT and CPRTFRSTT are \*DECOMPRESS.

Top

---

## SNA compression character (SNACPRCHR)

Specifies information about the SNA compression algorithm as it pertains to the requested object. If omitted, the implication is that SNA compression does not pertain to the requested object.

**\*BLANK**

When SNA compression is requested, it is optionally specified with the implied default being the (X'40') character. Otherwise, it is not specified.

**SNA-prime-character**

The prime compression character to be associated with the single control byte (SCB) used by the SNA compression algorithm. Valid values are '00'X - 'FF'X

**Note:** SNACPRCHR is valid only when ACTION(\*RTV), ACTION(\*SND), ACTION(\*SNDRUN) or ACTION(\*SNDINS) is specified.

Top

---

## User compression information (USRCPRINF)

Specifies information about a named user compression algorithm as it pertains to the requested object. If omitted, the implication is that user compression does not pertain to the requested object.

### *user-compression-name*

The name of the user compression algorithm that pertains to the requested object. It should be specified when user compression is requested. Otherwise, it is not specified.

### *user-parameters*

User parameters that apply to the user compression algorithm named in the user compression name. It is optionally specified when user compression is requested. Otherwise, it is not specified.

**Note:** USRCPRINF is valid only when ACTION(\*RTV), ACTION(\*SND), ACTION(\*SNDRUN) or ACTION(\*SNDINS) is specified.

Top

---

## Compression storage state (CPRSTGSTT)

Specifies whether or not the object should be stored in compressed format at the managed system once the object is sent.

### **\*NONE**

No compression storage state is specified.

### **\*DECOMPRESS**

Store the object in a decompressed format at the managed system. If the object is already decompressed when it arrives to the managed system, store it as received. Otherwise, decompress the object using the compression method used to catalog the object in the managed system before storing it.

### **\*COMPRESS**

Store the object in a compressed format at the managed system. When this value is specified, the compression algorithm must also be specified. If the object is already compressed when it arrives at the managed system, store the object as received. Otherwise, compress the object using the compression algorithm before storing it.

**Note:** CPRSTGSTT is valid only when ACTION(\*RTV), ACTION(\*SND), ACTION(\*SNDRUN) or ACTION(\*SNDINS) is specified.

Top

---

## Compression transfer state (CPRTFRSTT)

Specifies whether or not the object should be transferred to the managed system in compressed format.

### **\*NONE**

No compression transfer state is specified.

### **\*DECOMPRESS**

Transfer the object in decompressed format to the managed system. If already decompressed at

the source, transfer the object as stored. Otherwise decompress the object using the compression method used to catalog the object in the managed system before sending it.

#### **\*COMPRESS**

Compress the object using the compression algorithm and transfer the object in compressed format to the managed system. When this value is specified, the compression algorithm must also be specified. If the object is already compressed using a different algorithm at the source, decompress the object using the compression method used to catalog the object in the managed system before compressing and transferring it.

**Note:** CPRTFRSTT is valid only when ACTION(\*RTV), ACTION(\*SND), ACTION(\*SNDRUN) or ACTION(\*SNDINS) is specified.

Top

---

### **Activation use on install (ACTUSEINS)**

Specifies whether the component to be altered by the installation process will be trial version or production version. If activation use on install is \*TRIAL, it means the object should be installed in the trial area to be tested. Removability must then be \*YES.

#### **\*PROD**

Production version only.

#### **\*TRIAL**

Trial version only.

**Note:** ACTUSEINS is valid only when ACTION(\*INS) or ACTION(\*SNDINS) is specified.

Top

---

### **Alter active components (ALTACTIONCOMP)**

Specifies whether or not the managed system is allowed to apply the component alterations to the active system. If alterations cannot be applied, then such action is to be deferred until the next activation. This parameter is only allowed for \*SNDINS action.

#### **\*NONE**

No alter active component is specified.

#### **\*ALLOWED**

Managed system is allowed to apply the component alterations to the active system.

#### **\*NOTALLOWED**

Managed system is not allowed to apply the component alterations to the active system.

**Note:** ALTACTIONCOMP is valid only when ACTION(\*INS), ACTION(\*SNDINS), ACTION(\*RMV), or ACTION(\*UNINS) is specified.

Top

---

### **Auto accept (AUTOACP)**

Specifies whether the managed system accepts objects automatically if installation and any tests performed are successful, in order to release resources required to maintain removability as soon as possible. Like a separate accept request, the managed system deletes the objects after successful automatic acceptance.

**\*NONE**

No auto accept is specified.

**\*NO** Do not perform automatic acceptance.

**auto accept**

Auto acceptance possible values:

**\*YES** Perform automatic acceptance.

**\*DESIRED**

Perform automatic acceptance, if the specified managed system supports it.

The possible values are:

**delay** Specifies the number of days the entry point is expected to wait before accepting the object automatically.

**delay-days**

Valid range of days is 0-255.

**Note:** AUTOACP is valid only when ACTION(\*INS) or ACTION(\*SNDINS) is specified.

Top

---

## Auto remove (AUTORMV)

Specifies whether or not the managed system removes the object automatically if failure by either installation or a test.

**\*NONE**

No auto remove is specified.

**\*DESIRED**

Perform automatic removal if the specified managed system supports it.

**\*YES** Perform automatic removal.

**\*NO** Do not perform automatic removal.

**Note:** AUTORMV is valid only when ACTION(\*INS) or ACTION(\*SNDINS) is specified.

Top

---

## Pre test (PRETEST)

Specifies whether or not the entry point is to perform a test on the objects prior to installing them.

**\*DESIRED**

Perform a pretest if the specified managed system supports it.

**\*YES** Perform a pretest.

**\*NO** Do not perform a pretest.

**Note:** PRETEST is valid only when ACTION(\*INS) or ACTION(\*SNDINS) is specified.

Top



---

## Post test (POSTTEST)

Specifies whether or not the entry point is to perform a test on the objects after installing or removing them.

### \*DESIRED

Perform a posttest if the specified managed system supports it.

\*YES Perform a posttest.

\*NO Do not perform a posttest.

**Note:** POSTTEST is valid only when ACTION(\*INS), ACTION(\*SNDINS) or ACTION(\*RMV) is specified.

Top

---

## Allow remove (ALWRMV)

Specifies whether or not objects are to be installed in a removable manner (so that a subsequent remove action can be issued against the objects).

\*YES Install the object in a removable manner.

### \*DESIRED

Install the object in a removable manner if the specified managed system supports it.

\*NO Do not install the object in a removable manner.

**Note:** ALWRMV is valid only when ACTION(\*INS) or ACTION(\*SNDINS) is specified.

Top

---

## Initiate parameters (PARM)

Specifies the parameters to be passed to the program to be run.

### *parameter data*

Specifies the parameters to be passed when starting the program. Each parameter is a 1 to 253 character. The initial display allows 25 characters to be typed. By typing an & in position 1, the field expands to accommodate longer parameters. If the parameters include blanks or special characters, enclose them in apostrophes.

**Note:** PARM is valid only when ACTION(\*RUN) or ACTION(\*SNDRUN) is specified.

Top

---

## Keep object (KEEPOBJ)

Specifies whether or not the object should be kept or deleted after the function has been successfully performed.

### \*NONE

No keep object is specified.

\*YES The object should be kept after performing the function.

\*NO The object should be deleted after performing the function.

**Note:** KEEPOBJ is valid only when ACTION(\*RUN), ACTION(\*INS), ACTION(\*SNDRUN) or ACTION(\*SNDINS) is specified.

Top

---

## Corequisite change name list (COREQCHGNL)

Specifies a list of SNA/file services (SNA/FS) global names identifying the names of objects that are to be installed by the entry point as a part of the installation of the object to be retrieved. A maximum of six corequisite change names are allowed. The global name is a unique name you assign to the object so that it is not confused with any other object in a network. The global name represents the name that will be used to locate the appropriate catalog entry on both the central site and managed systems.

### \*NONE

No corequisite change name list is specified.

### *global-name-token*

One of a series of 1-16 character tokens that uniquely identify the object on which the action is to be performed. Characters A through Z and 0 through 9 may be used. Other special values (@, #, and \$) may be used for tokens that represent network IDs and system names.

**Note:** COREQCHGNL is valid only when ACTION(\*INS) or ACTION(\*SNDINS) is specified.

Top

---

## Replace object (REPLACE)

Specifies whether the object should be replaced if it already exists.

\*NO The object must be added.

\*YES The object must be replaced.

### \*ALLOWED

The object should be replaced or added.

**Note:** REPLACE is valid only when ACTION(\*RTV), ACTION(\*SND), ACTION(\*SNDRUN) or ACTION(\*SNDINS) is specified.

Top

---

## To be deleted name (TODLTNAME)

Specifies the name of the object, at the managed system location, that is to be deleted.

Special values in a token position indicate how to search for the object. By specifying \*ANY in a token position, the token is ignored when searching for the correct object.

### \*NONE

No to be deleted name is specified.

The special values allowed for the first token are: \*NETID, \*SERVER, \*TARGET, \*MDDATE, and \*MDTIME.

### Element 1: Token 1

The possible values are:

**\*NETID**

The first global name token value is a network ID generated by the command from the network attributes. The network ID is determined by the current value of the LCLNETID network attribute value.

**\*SERVER:**

This token is stored within the change request activity with the value &SERVER, and is replaced by the short name of the change control server when the object is deleted.

**\*TARGET:**

This token is stored within the change request activity with the value &TARGET, and is replaced by the short name of the target when the object is deleted.

**\*MDDATE:**

This token is stored within the change request activity with the value &DATE, and is replaced by the date the object was last modified.

**\*MDTIME:**

This token is stored within the change request activity with the value &TIME, and is replaced by the time the object was last modified.

***global-name-token-1***

Specify the first token of the global name. The first token is recommended to be the registered enterprise ID or network ID.

**Element 2-10: Token 2-10**

**\*ANY** Any token value matches when searching for the object where the action is to be performed.

**\*HIGHEST**

The object with the highest token value has the action performed on it. The token must be ordered. This is useful when a token in a global name is used to indicate a different version of the object and you need to manipulate the object with the highest version level.

**\*LOWEST**

The object with the lowest token value has the action performed on it. The token must be ordered. This is useful when a token in a global name is used to indicate a different version of the object and you need to manipulate the object with the lowest version level.

**\*NETID**

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

**\*CPNAME**

The control point name of this system is used. The control point name is determined by the current value of the LCLCPNAME network attribute value.

**\*SERVER:**

This token is stored within the change request activity with the value &SERVER, and is replaced by the short name of the change control server when the object is deleted.

**\*TARGET:**

This token is stored within the change request activity with the value &TARGET, and is replaced by the short name of the target when the object is deleted.

**\*MDDATE:**

This token is stored within the change request activity with the value &DATE, and is replaced when distributed by the date the object was last deleted.

**\*MDTIME:**

This token is stored within the change request activity with the value &TIME, and is replaced when distributed by the time the object was last deleted.

### *global-name-token-n*

Specify one of a series of 1 to 16 character tokens that uniquely identify the object on which the action is to be performed. Characters A through Z and 0 through 9 can be used. Other special values (@, #, and \$) can be used for tokens that represent network IDs and system names.

#### **Notes:**

1. TODLTNAME is valid only when ACTION(\*SND), ACTION(\*SNDRUN) or ACTION(\*SNDINS) is specified.
2. The number of tokens specified in the TODLTNAME parameter must match the number of tokens specified in the GLBNAME parameter.
3. TODLTNAME cannot be specified if the REPLACE value is \*NO.

Top

---

## **Refresh level (REFLVL)**

Specifies the level of the software component to be uninstalled.

### \*NONE

No refresh level is specified.

### *refresh level*

Specify a numeric character string of up to 16 digits.

**Note:** REFLVL is valid only when ACTION(\*UNINS) is specified.

Top

---

## **Force uninstall (FRCUNINS)**

Specifies whether to perform the uninstallation even if the software component includes an object that is waiting to be installed or an object that is currently being installed.

\*NO Do not allow uninstallation if there are objects, for the software component, waiting to be installed.

\*YES Allow uninstallation even if there are objects, for the software component, waiting to be installed.

**Note:** FRCUNINS is valid only when ACTION(\*UNINS) is specified.

Top

---

## **Start time on managed system (RMTSTRTIME)**

Specifies the date and time when the activity can begin running on the managed system. The current date and time values and next date values are determined when the activity begins running at the central site system based on the central site date and time.

### **Single value:**

### \*NONE

No start time on the managed system is specified.

### **Element 1: Time Zone**

The time zone of the remote start time.

The possible values are:

**\*LCLSYS**

The remote start time is specified in the time zone of the central site system. The start time is converted to Greenwich Mean Time.

**\*MGDSYS**

The remote start time is specified in the time zone of the managed system.

**Element 2: Start After Time**

This is the definition of the time after which the activity is to start.

**\*CURRENT**

This function can start on the managed system at any time on or after the time this activity is started on the central site system on the date specified in element 2.

*start-after-time*

Specify the time when this function can start on the managed system. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

**Element 3: Start After Date**

**\*CURRENT**

This function starts on the managed system on any date on or after the activity starts on the central site system.

**\*NEXT**

This function starts on the managed system on any date after the activity starts on the central site system.

*start-after-date*

Specify the date after the functions start on the managed system. The date must be specified in the job date format.

**Notes:**

1. The special values \*CURRENT and \*NEXT cannot be specified for the date and the time when the time zone value \*MGDSYS is specified.
2. This parameter is valid only when these actions are specified: \*ACP, \*ACT, \*RUN, \*INS, \*RMV, \*SNDRUN, \*SNDINS, or \*UNINS.

Top

---

## Activity conditions (COND)

Specifies which conditions must be met before this activity can be performed. Each condition identifies an activity that must run before this activity and the value the end code from that activity must have to allow this activity to run. The default condition is that the previous activity (in alphabetical order) must complete successfully before this activity can be run.

**Element 1: Conditioning Activity**

The activity that must be run before this activity.

The possible values are:

**\*PRV**

This activity is conditioned on the previous activity. Activities are ordered alphabetically by activity name. If the activity being added is the first activity, a previous activity does not exist and any condition with \*PRV is marked as having been met.

### *conditioning-activity-name*

Specify the name of the activity that must be run before this activity. The activity name specified in the activity (ACTIVITY) parameter cannot be specified in the conditioning activity name. An activity cannot be conditioned on itself.

### *generic\*-conditioning-activity-name*

Specify the generic name of the activities that must be run before this activity.

#### **Element 2: Relational Operator**

This element is the relational operator to use when comparing the end code from the conditioning activity.

- \*EQ** Equal
- \*GT** Greater than
- \*LT** Less than
- \*NE** Not equal
- \*GE** Greater than or equal
- \*LE** Less than or equal

#### **Element 3: Condition Code**

This element is the value compared to the actual end code of the conditioning activity.

### **\*SUCCESS**

The activity ended successfully (0 <= end code <= 9). This end code can only be specified with relational operator \*EQ or \*NE.

**\*FAIL** The activity failed (10 <= end code <= 89). This end code can only be specified with relational operator \*EQ or \*NE.

### **\*NOTRUN**

The activity never started (90 <= end code <= 99). This end code is only specified with relational operator \*EQ or \*NE.

**\*ANY** The activity ended with any end code. This end code is only specified with relational operator \*EQ.

### *end-code*

Specify an integer value (0-99) that indicates the result of an activity (success or failure). The end code ranges and descriptions are:

- 00** Activity completed successfully.
- 01-09** Activity completed with warning messages.
- 10-29** Activity did not complete successfully.
- 30-39** Activity was canceled by a user before it completed.
  - 30 = Activity ended with \*CNTRLD option
  - 35 = Activity ended with \*IMMED option
  - 39 = Activity ended with \*FRCFAIL option
- 40-49** Activity was not run due errors detected by the application.
  - 40 = Activity not run for security reasons
- 90-99** Activity was not run because conditions or schedules were not met.
  - 95 = Scheduled start time expired
  - 99 = Conditions cannot be met

#### Element 4: Condition Mode

This element indicates which systems the conditioning activity must have completed on before this activity can be performed.

##### \*ALLNODES

The conditioning activity specified must complete on all nodes before this activity runs.

##### \*SAMENODE

When the conditioning activity specified completes for a given node, the activity specified on the ACTIVITY parameter may run for that same node even though the conditioning activity specified cannot have completed for all other nodes. In the case where this activity lists a node not in the conditioning activity, this activity may run for that node; the condition is ignored.

##### \*NONE

There are no conditions for this activity.

Top

---

## Activity start time (STRTIME)

Specifies the date and time when this activity can be started on the central site system. The current date and time values, and next date values are determined when the change request is submitted.

#### Element 1: Start After Time

The possible values are:

##### \*CURRENT

This activity can start any time on or after the time when the change request is submitted.

##### *start-after-time*

Specify the time when this activity can start. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

#### Element 2: Start After Date

##### \*CURRENT

This activity can start on or after the date on which the change request is submitted.

##### \*NEXT

The activity can start on any date after the date when the change request is submitted.

##### *start-after-date*

Specify the date after which this activity can start. The date must be specified in the job date format.

#### Element 3: Start Before Time

This element is ignored if the start before date is \*ANY.

\*ANY The activity can start at any time on or before the start before date.

##### \*CURRENT

The activity must start before the time when the change request was submitted on the date specified on the start before data element.

### *start-before-time*

Specify the time before the activity must be started. If the activity cannot be started before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

#### **Element 4: Start Before Date**

**\*ANY** The activity can start at any time after the start after time and the start after date.

#### **\*CURRENT**

The activity must start on the date the change request is submitted.

#### **\*NEXT**

The activity must start by the day after the date the change request is submitted.

### *start-before-date*

Specify the date before the activity must start. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

Top

---

## **Hold activity (HOLD)**

Hold the activity when the change request is submitted.

**\*NO** The activity is not held and will run when all conditions are met.

**\*YES** The activity is held for all nodes when the change request is submitted. It must be released by you before it runs.

Top

---

## **Examples**

The following examples illustrate how to use the ADDCRQA command to schedule activities to be performed on managed systems controlled by a NetView Distribution Manager/6000 change control server. The examples shown here are grouped according to their activity action:

- Accept
- Activate
- Delete
- Run
- Install
- Remove
- Retrieve
- Send, send and run, send and install
- Uninstall

### **Example 1: Accept Actions**

The examples shown here match those shown for the install examples and demonstrate how some of the installed objects can be accepted.

- The install activity added to the change request description is:



```

ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT01)
          ACTION(*INS)
          GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          ACTUSEINS(*TRIAL)  ALWRMV(*YES)

```

When the change request is submitted and if the activity is run successfully, object IBM 1234567 PMGRAB UPD 1 2 is installed (to be tested) in the trial area of managed system MARYPWS1. MARYPWS1 is controlled by the change control server ROMSERV1. Because the object is installed in the trial area, the object cannot be accepted.

- The install activity added to the change request description is:

```

ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT02)
          ACTION(*INS)
          GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          ACTUSEINS(*PROD)  ALTACTCOMP(*ALLOWED)
          ALWRMV(*NO)
          RMTSTRTIME((*MGDSYS) (8:30:00 12/25/02))

```

When the change request is submitted and if the activity is successfully run, object IBM 1234567 PMGRAB UPD 1 2 is installed in the active area of the system MARYPWS1. MARYPWS1 is managed by the change control server ROMSERV1. The object is installed in a nonremovable manner, and thus, the object does not need to be accepted.

- The install activity added to the change request description is:

```

ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT03)
          ACTION(*INS)
          GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          COREQCHGNL((REXX PROC UPDATE CONFIG))
          ACTUSEINS(*PROD)
          ALTACTCOMP(*NOTALLOWED)  AUTOACP(*YES)
          ALWRMV(*YES)

```

The objects are automatically accepted with a required activation if the following conditions are met:

- The change request is submitted
- The activity runs successfully
- The installation completes successfully (the object IBM 1234567 PMGRAB UPD 1 2 and its corequisite REXX PROC UPDATE CONFIG were installed successfully)

If the installation did not complete successfully, the objects IBM 1234567 PMGRAB UPD 1 2 and REXX PROC UPDATE CONFIG are not automatically accepted. You can accept the objects running the following activities:

```

ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(ACCACT01)
          ACTION(*ACP)
          GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          COND((INSACT03 *EQ 20 *SAMENODE))
ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(ACCACT02)
          ACTION(*ACP)
          GLBNAME(REXX PROC UPDATE CONFIG)
          CPNAME((ROMSERV1 MARYPWS1))
          COND((INSACT03 *EQ 20 *SAMENODE))

```

- The install activity added to the change request description is:

```

ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT04)
          ACTION(*INS)
          GLBNAME(EURO WORDPROD UPD 2 3 US)
          CPNAME((EUROITAL FREDSSWS))
          ALWRMV(*YES)

```

When the change request is submitted and if the activity is run successfully, an object is installed in a removable manner. The object can be accepted adding an activity to accept the previously installed object EURO WORDPROD UPD 2 3 US on managed system FREDSSWS. Managed system FREDSSWS is controlled by the change control server EUROITAL.

```

ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(ACCACT03)
          ACTION(*ACP)  GLBNAME(EURO WORDPROD UPD 2 3 US)
          CPNAME((EUROITAL FREDSW))
          COND((INSACT04 *EQ *SUCCESS *SAMENODE))

```

### Example 2: Activate Actions

- It is assumed that an install activity was issued as follows:

```

ADDCRQA  CRQD(CCLIB/CRQINSACT)  ACTIVITY(INSACT01)
          ACTION(*INS)  GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          ACTUSEINS(*PROD)  ALTACTCOMP(*NOTALLOWED)
          AUTOACP(*YES)
          COREQCHGNL((REXX PROC UPDATE CONFIG))

```

- When the change request is submitted and if the activity is run successfully, objects IBM 1234567 PMGRAB UPD 1 2 and REXX PROC UPDATE CONFIG are automatically accepted. To apply the component alterations to the active system, an activation is required. To activate previously installed changes, run an activity like the following:

```

ADDCRQA  CRQD(CCLIB/CRQINSACT)  ACTIVITY(ACTACT01)
          ACTION(*ACT)  CPNAME((ROMSERV1 MARYPWS1))
          ACTFRC(*NO *HOURS 6)  ACTUSEACT(*PROD)
          STRTIME((2:00:00 *NEXT) (8:00:00 *NEXT))

```

The activation will occur after 2 a.m. but before 8 a.m. the next morning the request is submitted. The MARYPWS1 system waits up to six hours before proceeding with the activation if the managed system is still active.

- Add an activity to schedule the activation of all previously installed objects on managed system FREDSW. FREDSW is controlled by the change control server EUROITAL. The activation occurs as soon as possible after 11 p.m. on 15 April 2002 in the time zone where FREDSW is located. The activation occurs even if the managed system FREDSW is still active.

```

ADDCRQA  CRQD(CCLIB/CRQINSACT)  ACTIVITY(ACTACT02)
          ACTION(*ACT)  CPNAME((EUROITAL FREDSW))
          ACTFRC(*YES)
          RMTSTRTIME(*MGDSYS (23:00:00 04/15/02))

```

### Example 3: Delete Actions

- Add an activity to delete the object EURO WORDPROD UPD 2 3 US on managed system FREDSW. FREDSW is controlled by the change control server EUROITAL.

```

ADDCRQA  CRQD(CCLIB/CRQDLT)  ACTIVITY(DLTACT01)
          ACTION(*DLT)
          GLBNAME(EURO WORDPROD UPD 2 3 US)
          CPNAME((EUROITAL FREDSW))

```

- Add an activity to delete all test files from systems MARYPWS1, MARYPWS2, MARYPWS3, and MARYPWS4. These systems are controlled by the change control server ROMSERV1. The test files on these systems are cataloged as EURO SPELLCHECK TEST file-name.

```

ADDCRQA  CRQD(CCLIB/CRQSNDLT)  ACTIVITY(DLTACT01)
          ACTION(*DLT)  GLBNAME(EURO SPELLCHECK TEST *ANY)
          CPNAME((ROMSERV1 MARYPWS1) (ROMSERV1 MARYPWS2)
                 (ROMSERV1 MARYPWS3) (ROMSERV1 MARYPWS4))
          COND((SNDACT01 *EQ *SUCCESS *ALLNODES))

```

The delete activity is performed only if the object EURO SPELLCHECK EXE 1 US is successfully sent to those systems:

```

ADDCRQA  CRQD(CCLIB/CRQSNDLT)  ACTIVITY(SNDACT01)
          ACTION(*SND)  GLBNAME(EURO SPELLCHECK EXE 1 US)
          CPNAME((ROMSERV1 MARYPWS1) (ROMSERV1 MARYPWS2)
                 (ROMSERV1 MARYPWS3) (ROMSERV1 MARYPWS4))
          REPLACE(*ALLOWED)

```

### Example 4: Run Actions

- Add an activity to run the program or script known by the global name EURO VIRUSCHK EXE 1 US on managed system FREDWS. Pass the parameter /usr/bin to the program or script. The activity is run as soon as possible.

```
ADDCRQA  CRQD(CCLIB/CRQRUN)  ACTIVITY(RUNACT01)
          ACTION(*RUN)
          GLBNAME(EURO VIRUSCHK EXE 1 US)
          CPNAME((EUROITAL FREDWS))  PARM("/usr/bin")
```

- Add an activity to run the program known by the global name EURO WORDPROC EXE 2 US on managed system JOHNSWS. This activity should be run immediately by the PS/2. The object should be kept in the NetView/DM2 catalog. The change control server is in the same network as the local system.

```
ADDCRQA  CRQD(CCLIB/CRQRUN)  ACTIVITY(RUNACT02)
          ACTION(*RUN)
          GLBNAME(EURO WORDPROC EXE 2 US)
          CPNAME((*NETATR JOHNSWS))  KEEPOBJ(*YES)
```

### Example 5: Install Actions

The first four examples shown here match those shown for the accept and remove examples.

- Add an activity to install the object identified by the global name IBM 1234567 PMGRAB UPD 1 2 on the change control client machine (managed system) called MARYPWS1. MARYPWS1 is controlled by the change control server ROMSERV1. The object is to be installed in the trial area.

```
ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT01)
          ACTION(*INS)  GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          ACTUSEINS(*TRIAL)  ALWRMV(*YES)
```

- Add an activity to install the object identified by the global name IBM 1234567 PMGRAB UPD 1 2 on the change control client machine (managed system) called MARYPWS1. MARYPWS1 is controlled by the change control server ROMSERV1. The object is to be installed permanently in the active area. The activity is to be processed at 8:30 a.m. on 25 December 2002.

```
ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT02)
          ACTION(*INS)  GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          ACTUSEINS(*PROD)  ALTACTCOMP(*ALLOWED)
          ALWRMV(*NO)
          RMTSTRTIME((*MGDSYS) (8:30:00 12/25/02))
```

- Add an activity to install the objects identified by the global names IBM 1234567 PMGRAB UPD 1 2 and its corequisite REXX PROC UPDATE CONFIG on the change control client (managed system) MARYPWS1. MARYPWS1 is controlled by the change control server ROMSERV1. The objects are to be installed in a nonremovable manner and are to be automatically accepted. An activation is required to apply the component alterations to the active system. The activity is to be processed immediately. If the installation of one of the objects fails, the other installation fails also.

```
ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT03)
          ACTION(*INS)  GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          ACTUSEINS(*PROD)  ALTACTCOMP(*NOTALLOWED)
          AUTOACP(*YES)  ALWRMV(*YES)
          COREQCHGNL((REXX PROC UPDATE CONFIG))
```

- Add an activity to install the object identified by the global name EURO WORDPROC UPD 2 3 US on the change control client machine (managed system) called FREDWS. FREDWS is controlled by the change control server EUROITAL. The object is to be installed in a removable manner.

```
ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT04)
          ACTION(*INS)  GLBNAME(EURO WORDPROC UPD 2 3 US)
          CPNAME((EUROITAL FREDWS))
          ALWRMV(*YES)
```

- Add an activity to schedule the install of the objects EURO.WORDPROC.REF.2.US and EURO.WORDPROC.UPD.2.3.US on managed system FREDWS. FREDWS is controlled by the change

control server EUROITAL. The object is installed in the active area in a removable manner. The installation is not automatically accepted. The installation is scheduled for 3 p.m. on 1 January 2003.

```
ADDCRQA  CRQD(CCLIB/CRQRTVINS)  ACTIVITY(INSACT05)
          ACTION(*INS)  GLBNAME(EURO WORDPROC REF 2 US)
          CPNAME((EUROITAL FREDSW))
          ALTACTCOMP(*ALLOWED)
          COREQCHGNL(EURO WORDPROC UPD 2 3 US)
          RMTSTRTIME(*MGDSYS (15:00:00 1/01/03))
          COND((RTVACT01 *EQ *SUCCESS *ALLNODES))
```

The activity can be performed only if the object EURO WORDPROC UPD 2 3 US is successfully retrieved from the change control client BRIGSW.

```
ADDCRQA  CRQD(CCLIB/CR7)  ACTIVITY(RTVACT01)  ACTION(*RTV)
          GLBNAME(EURO WORDPROC UPD 2 3 US)
          CPNAME((EUROITAL BRIGSW))
          STRTIME((22:00:00 12/31/02) (06:00:00 1/01/03))
```

- Add an activity to install the object identified by the global name IBM 1234567 WINDMB UPD 1 2 on the production area of managed system MARYPWS1. If the installation fails, the object is removed automatically.

```
ADDCRQA  CRQD(CCLIB/CRQINS)  ACTIVITY(INSACT06)
          ACTION(*INS)  GLBNAME(IBM 1234567 WINDMB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          ACTUSEINS(*PROD)  AUTORMV(*YES)  ALWRMV(*YES)
```

### Example 6: Remove Actions

The examples shown here match those shown for the install examples and demonstrate how the installed objects can be removed.

- The install activity added to the change request description is:

```
ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT01)
          ACTION(*INS)  GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          ACTUSEINS(*TRIAL)  ALWRMV(*YES)
```

When the change request is submitted and if the activity is run successfully, object IBM 1234567 PMGRAB UPD 1 2 is installed (to be tested) in the trial area of managed system MARYPWS1. MARYPWS1 is controlled by the change control server ROMSERV1. The object IBM 1234567 PMGRAB UPD 1 2 can be removed when you run the following activity:

```
ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(RMVACT01)
          ACTION(*RMV)  GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))  POSTST(*NO)
```

A posttest is not required.

- The install activity added to the change request description is:

```
ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT02)
          ACTION(*INS)  GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          ACTUSEINS(*PROD)  ALTACTCOMP(*ALLOWED)
          ALWRMV(*NO)
          RMTSTRTIME((MGDSYS) (8:30:00 12/25/02))
```

When the change request is submitted and if the activity is run successfully, object IBM 1234567 PMGRAB UPD 1 2 is installed in the active area of the system MARYPWS1. MARYPWS1 is managed by the change control server ROMSERV1. The object is installed in a nonremovable manner.

- The install activity added to the change request description is:

```
ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT03)
          ACTION(*INS)  GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          ACTUSEINS(*PROD)  ALTACTCOMP(*NOTALLOWED)
          AUTOACP(*YES)  ALWRMV(*YES)
          COREQCHGNL((REXX PROC UPDATE CONFIG))
```

- The objects are automatically accepted if the following conditions are met:

- The change request is submitted
- The activity is run
- The installation completes successfully (the object IBM 1234567 PMGRAB UPD 1 2 and its corequisite REXX PROC UPDATE CONFIG were installed successfully)

All objects are installed in a removable manner, but an activation is required. The objects can be removed by adding and running activities like the following:

```
ADDCRQA CRQD(CCLIB/CRQACPRMV) ACTIVITY(RMVACT02)
        ACTION(*RMV) GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
        CPNAME((ROMSERV1 MARYPWS1))
        COND((INSACT03 *EQ *SUCCESS *SAMENODE))
ADDCRQA CRQD(CCLIB/CRQACPRMV) ACTIVITY(RMVACT03)
        ACTION(*RMV) GLBNAME(REXX PROC UPDATE CONFIG)
        CPNAME((ROMSERV1 MARYPWS1))
        COND((INSACT03 *EQ *SUCCESS *SAMENODE))
```

- The install activity added to the change request description is:

```
ADDCRQA CRQD(CCLIB/CRQACPRMV) ACTIVITY(INSACT04)
        ACTION(*INS) GLBNAME(EURO WORDPROD UPD 2 3 US)
        CPNAME((EUROITAL FREDSW)) ALWRMV(*YES)
```

When the change request is submitted and if the activity is run successfully, the object is installed in a removable manner. The object can be removed when you add an activity to remove the previously installed object EURO WORDPROD UPD 2 3 US on managed system FREDSW. FREDSW is controlled by the change control server EUROITAL.

```
ADDCRQA CRQD(CCLIB/CRQINSRMV) ACTIVITY(RMVACT04)
        ACTION(*RMV) GLBNAME(EURO WORDPROD UPD 2 3 US)
        CPNAME((EUROITAL FREDSW))
        COND((INSACT04 *EQ *SUCCESS *SAMENODE))
```

### Example 7: Retrieve Actions

- Add an activity that causes the file EURO WORDPROC UPD 2 3 US to be retrieved from the managed system JOHNSWS. The file is transferred in a compressed format and then stored in a decompressed format. The SNA compression type is used.

```
ADDCRQA CRQD(CCLIB/CRQRTV) ACTIVITY(RTVACT01)
        ACTION(*RTV) GLBNAME(EURO WORDPROC UPD 2 3 US)
        CPNAME((EUROITAL JOHNSWS)) CPRTYPE(*SNA)
        CPRSTGTT(*DECOMPRESS) CPRTFRSTT(*COMPRESS)
        REPLACE(*ALLOWED)
```

- Add an activity that causes the object EURO PCSOFT UPD 2 3 US to be retrieved from the managed system JOHNSWS. The file is transferred and stored in a compressed format. The user compression name NVDMLZW is used. This user compression type is supported if the object is not decompressed at the central site system.

```
ADDCRQA CRQD(CCLIB/CRQRTV) ACTIVITY(RTVACT02)
        ACTION(*RTV) GLBNAME(EURO PCSOFT UPD 2 3 US)
        CPNAME((EUROITAL JOHNSWS)) CPRTYPE(*USER)
        CPRSTGTT(*COMPRESS) CPRTFRSTT(*COMPRESS)
        USRCPRINF(NVDMLZW 37 '/D') REPLACE(*ALLOWED)
```

- Add an activity to retrieve the most recent sales file from each system. The files are cataloged as EURO SALES system-name date-created. The file is sent and stored in a compressed format. The SNA compression algorithm is used. The file is retrieved after 10 p.m. on the day the request is submitted but before 6 a.m. the next morning when the stores open. All files retrieved are added to the distribution repository when they are retrieved.

```
ADDCRQA CRQD(CCLIB/CRQRTV) ACTIVITY(RTVACT03)
        ACTION(*RTV) GLBNAME(EURO SALES *ANY *HIGHEST)
        NODL(STORES) CPRTYPE(*SNA) CPRSTGTT(*COMPRESS)
        CPRTFRSTT(*COMPRESS) REPLACE(*NO)
        STRTIME((22:00:00 *CURRENT) (06:00:00 *NEXT))
```

### Example 8: Send, Send and Run, and Send and Install Actions

- Add an activity to send the file EURO SPELLCHECK EXE 1 US to the managed system FREDSWWS. FREDSWWS is controlled by the change control server EUROITAL. The file is not compressed. The file is replaced or added at the managed system.

```
ADDCRQA  CRQD(CCLIB/CRQSND)  ACTIVITY(SNDCT01)
          ACTION(*SND)  GLBNAME(EURO SPELLCHECK EXE 1 US)
          CPNAME((EUROITAL FREDSWWS))  REPLACE(*ALLOWED)
```

- Add an activity to send and install the two objects EURO WORDPROC REF 2 US and EURO WORDPROC UPD 2 3 US on managed system FREDSWWS with these specified attributes. The installation is done to the active area in a removable manner. The installation is not automatically accepted. It is scheduled for 3 a.m. on 13 April 2002 in the time zone where the managed system is located.

```
ADDCRQA  CRQD(CCLIB/CRQSND)  ACTIVITY(SNDCT02)
          ACTION(*SNDINS)  GLBNAME(EURO WORDPROC REF 2 US)
          CPNAME((EUROITAL FREDSWWS))  ALTACTCOMP(*ALLOWED)
          COREQCHGNL((EURO WORDPROC UPD 2 3 US))
          RMTSTRTIME(*MGDSYS (15:00:00 04/13/02))
```

- Add an activity to send and run the program known by the global name EURO.VIRUSCHK.EXE.1.US on managed system FREDSWWS. Pass the parameter /usr/bin to the program. The program is run as soon as possible.

```
ADDCRQA  CRQD(CCLIB/CRQSND)  ACTIVITY(SNDCT03)
          ACTION(*SNDRUN)  GLBNAME(EURO VIRUSCHK EXE 1 US)
          CPNAME((EUROITAL FREDSWWS))  PARM("/usr/bin")
```

- Add activities to retrieve the program identified by the global name CUSTNET PCSOFT WDWAPP VER3 941128 from the PS/2 DEVPS2, which is controlled by the central site CUSTNET. Send and run on all of the PS/2s in the southeast area. The central site system runs the program at 11 p.m. in the time zone where the PS/2 is located only if the retrieve from the PS/2 DEVPS2 was successful. The activity names are generated.

```
ADDCRQA  CRQD(CCLIB/CRQRTVSND)  ACTIVITY(*GEN)
          ACTION(*RTV)
          GLBNAME(CUSTNET PCSOFT WDWAPP VER5 021230)
          CPNAME((CUSTNET DEVPS2))
```

```
ADDCRQA  CRQD(CCLIB/CRQRTVSND)  ACTIVITY(*GEN)
          ACTION(*SNDRUN)
          GLBNAME(CUSTNET PCSOFT WDWAPP VER5 021230)
          NODL(PS2SE)
          RMTSTRTIME(*MGDSYS (23:00:00 12/30/02))
          COND(*PRV *EQ *SUCCES *SAMENODE)
```

### Example 9: Uninstall Actions

- Add an activity to uninstall the component EURO WORDPROC from managed system FREDSWWS at the time and date specified. The system must be in a state of inactivity at that time. Do not perform the uninstall action if there are objects waiting to be installed.

```
ADDCRQA  CRQD(CCLIB/CRQUNINS)  ACTIVITY(UNSACT01)
          ACTION(*UNINS)  COMPNAME(EURO WORDPROC)
          CPNAME((EUROITAL FREDSWWS))
          ALTACTCOMP(*NOTALLOWED)  FRCUNINS(*NO)
          REFLVL("2")
          RMTSTRTIME(*MGDSYS (14:00:00 05/20/02))
```

Top

---

## Error messages

None

Top



## Add License CRQ Activity (ADDLICCRQA)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Add License CRQ Activity (ADDLICCRQA) command adds a license key distribution activity to a change request description that performs a license key distribution function.

The activity can be conditioned so that it will only run after one or more other activities have completed (successfully or unsuccessfully). The activity can also be scheduled to run at a date and time in the future.

### Restrictions:

1. This command is shipped with public \*EXCLUDE authority.
2. The user must have \*CHANGE authority to the change request description object and \*EXECUTE authority to the library.
3. The user adding the activity does not need to be authorized to any objects that are to be manipulated.
4. If a node list (NODL) value is specified, the node list can only contain entries that have a value of \*SNA for the address type.
5. The license key information does not need to exist in the license repository when a license activity is being added to a change request, but it must exist by the time the activity runs.

### Notes:

1. All conditions must be satisfied before the activity can be performed.
2. The start times indicate when the activity can be started. Actual start times may be later due to network and system delays.
3. All the existing records in the central site license repository containing the license key information for the specified product are sent to the specified managed system or systems. Only those ones that match the system serial number are added to the managed system license repository.

Top

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACTIVITY	Activity	<i>Name, *GEN, *LAST</i>	Optional, Positional 2
ACTION	Action	*SND	Optional, Positional 3
PRDID	Product ID	<i>Character value, *ALL</i>	Optional, Positional 4
LICTRM	License term	<i>Character value, *ALL</i>	Optional, Positional 5

Keyword	Description	Choices	Notes
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: <b>*NONE</b> Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CPNAME	Managed system node names	Single values: <b>*NONE</b> Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Communications name</i>	
TEXT	Text 'description'	<i>Character value, *GEN, *BLANK</i>	Optional
COND	Activity conditions	Single values: <b>*NONE</b> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Activity	<i>Generic name, name, *PRV</i>	
	Element 2: Relational operator	<b>*EQ, *GT, *LT, *LE, *NE, *GE</b>	
	Element 3: End code	0-99, <b>*SUCCESS, *FAIL, *NOTRUN, *ANY</b>	
	Element 4: Condition mode	<b>*ALLNODES, *SAMENODE</b>	
STRTIME	Activity start time	<i>Element list</i>	Optional
	Element 1: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time, *CURRENT</i>	
	Element 2: Date	<i>Date, *CURRENT, *NEXT</i>	
	Element 2: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time, *ANY, *CURRENT</i>	
	Element 2: Date	<i>Date, *ANY, *CURRENT, *NEXT</i>	
	HOLD	Hold activity	

Top

---

## Change request description (CRQD)

Specifies the change request description object name.

The possible library values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched.

**\*CURLIB**

The current library for the job is used to locate the object.

*library-name*

Only the library named in this parameter is searched.

The possible change request description value is:

*change-request-description*

The name of the change request description object.

Top



---

## Activity (ACTIVITY)

Specifies the name of the activity to add to the change request description.

**\*GEN** An activity identifier will be generated. The activity ID is in the form QACTxxxxxx where xxxxxx is a number that is incremented for each activity added.

**\*LAST** The activity will be the last to run in the change request. .

### *activity-name*

A 10-character activity identifier.

Top

---

## Action (ACTION)

Specifies the license distribution function that is to be performed.

**\*SND** Send the license keys of the specified product and license term to the specified managed systems.

Top

---

## Product ID (PRDID)

Specifies the 7-character identifier of the product for which the license key information will be distributed.

**\*ALL** The license keys for all the products existing in the central site license repository will be distributed.

### *Product-Identifier*

Specifies the 7-character identifier of the product for which the license key information will be distributed.

Top

---

## License term (LICTRM)

Specifies the license term for which the license keys information will be sent.

**\*ALL** The license keys for all license term of the specified product existing in the focal point license repository will be distributed.

### *license-term*

Specify the license term in one of the following formats:

**Vx** The authorized usage limit is valid for the entire version of the product or feature.

**VxRy** The authorized usage limit is valid for the entire release of the product or feature.

### **VxRyMz**

The authorized usage limit is valid only for a modification of the product.

where "x" and "y" can be a number from 0 through 9, and "z" can be a number 0 through 9 or a letter A through Z.

Top

---

## Managed systems node list (NODL)

Specifies that the node list parameter is the object name that contains a list of systems that are the destinations for the activity. This parameter cannot be specified if the control point name (CPNAME) parameter is specified.

The possible value is:

### \*NONE

The systems on which this activity is to be performed are not identified by a node list. Individual control point names must be specified.

The possible library values are:

\*LIBL All of the libraries in the user and system portions of the job's library list are searched for the node list object.

### \*CURLIB

The current library for the job is used to locate the node list object.

### *library-name.*

Specify the name of the library to be searched.

The possible node list name value is:

### *node-list-name*

Specify the node list object name containing the list of systems on which the activity is to be performed.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if the node list (NODL) parameter is specified.

The possible values are:

### \*NONE

The systems on which this activity is to be performed are not identified individually. A node list must be specified.

### \*NETATR

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

### *network-identifier*

Specify the APPN network identifier of the managed system on which the activity is to be performed.

The possible control point value is:

### *control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed.

Top

---

## Text 'description' (TEXT)

Specifies the activity description.

**\*GEN** A text description is generated based upon the action chosen.

### *text-description*

Specify a 50-character description of the activity.

Top

---

## Activity conditions (COND)

Specifies which conditions must be met before this activity can be performed. Each condition identifies an activity that must run before this activity and the value the end code from that activity must have to allow this activity to run. The default condition is that the previous activity (in alphabetical order) must complete successfully before this activity can be run.

The possible conditioning activity values are:

The activity that must be run before this activity.

**\*PRV** This activity is conditioned on the previous activity. Activities are ordered alphabetically by activity name. If the activity being added is the first activity, a previous activity does not exist and any condition with \*PRV is marked as having been met.

### *conditioning-activity-name*

Specify the name of this activity that must run before this activity. The activity name specified in the activity (ACTIVITY) parameter cannot be specified in the conditioning activity name. An activity cannot be conditioned on itself.

### *generic\*-conditioning-activity-name*

Specify the generic name of the activities that must run before this activity.

The possible relational operator value is:

This element is the relational operator to use when comparing the end code from the conditioning activity.

**\*EQ** Equal

**\*GT** Greater than

**\*LT** Less than

**\*NE** Not equal

**\*GE** Greater than or equal

**\*LE** Less than or equal

The possible condition code value is:

This element is the value compared to the actual end code of the conditioning activity.

### **\*SUCCESS**

The activity ended successfully (0 <= end code <= 9). This end code can only be specified with relational operator \*EQ or \*NE.

**\*FAIL** The activity failed (10 <= end code <= 89). This end code can only be specified with relational operator \*EQ or \*NE.

**\*NOTRUN**

The activity never started (90 <= end code <= 99). This end code is only specified with relational operator \*EQ or \*NE.

**\*ANY** The activity ended with any end code. This end code is only specified with relational operator \*EQ.

**end-code**

Specify an integer value (0-99) that indicates the result of an activity (success or failure). The end code ranges and descriptions are:

- 00** Activity completed successfully.
- 01-09** Activity completed with warning messages.
- 10-29** Activity did not complete successfully.
- 30-39** Activity was canceled by a user before it completed.
  - 30 = Activity ended with \*CNTRLD option
  - 35 = Activity ended with \*IMMED option
  - 39 = Activity ended with \*FRCFAIL option
- 40-49** Activity was not run due to errors detected by the application.
  - 40 = Activity not run for security reasons
- 90-99** Activity was not run because conditions or schedules were not met.
  - 95 = Scheduled start time expired
  - 99 = Conditions cannot be met

The possible condition mode values are:

This element indicates which systems the conditioning activity must have completed on before this activity can be performed.

**\*ALLNODES**

The conditioning activity specified must complete on all nodes before this activity can run.

**\*SAMENODE**

When the conditioning activity specified completes for a given node, the activity specified on the ACTIVITY parameter may run for that same node even though the conditioning activity specified may not have completed on all other nodes. In the case where this activity lists a node not in the conditioning activity, this activity may run for that node; the condition is ignored.

The possible value is:

**\*NONE**

There are no conditions for this activity.

Top

---

## Activity start time (STRTIME)

Specifies the date and time when this activity can be started on the central site system. This is the date and time at which the specified license information will be sent. The current date and time values and next date values are determined when the change request is submitted.

The possible start after time values are:

**\*CURRENT**

This activity can start any time on or after the time when the change request is submitted.

***start-after-time***

Specify the time when this activity can start. The time can be entered as 4 or 6 digits (hhmm or hhmmss), where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). Without a time separator, specify a string of 4 to 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss) where the time separator specified for your job is used to separate the hours, minutes, and seconds.

The possible start after date values are:

**\*CURRENT**

This activity can start on or after the date on which the change request is submitted.

**\*NEXT**

The activity can start on any date after the date the change request is submitted.

***start-after-date***

Specify the date after this activity can start. The date must be specified in the job date format.

The possible start before time values are:

This element is ignored if the start before date is \*ANY.

**\*ANY** The activity can start at any time on or before the start before date.

**\*CURRENT**

The activity must start before the time at which the change request was submitted on the date specified on the start before date element.

***start-before-time***

Specify the time before which the activity must start. If the activity cannot be started before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional.

The time can be specified with or without a time separator such as a colon (:). Without a time separator, specify a string of 4 to 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss) where the time separator specified for your job is used to separate the hours, minutes, and seconds.

The possible start before date values are:

**\*ANY** The activity can start at any time after the start after time and the start after date.

**\*CURRENT**

The activity must start on the date the change request is submitted.

**\*NEXT**

The activity must start by the day after the date the change request is submitted.

***start-before-date***

Specify the date before the activity must start. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

Top

---

## Hold activity (HOLD)

Specifies that the activity be held when the change request is submitted.

**\*NO** The activity is not held. It runs when all conditions are met at the start time.

**\*YES** The activity is held for all nodes when the change request is submitted. It must be released by you before it runs.

Top

---

## Examples

### Example 1: Adding an Activity to Send Licenses

```
ADDLICCRQA CRQD(MYLIB/CR1) ACTIVITY(ACT01) ACTION(*SND)
           PRDID(1ACCOUN) LICTRM(V5R2M0)
           CPNAME((*NETATR SYS1))
```

Add an activity to send the licenses for the 1ACCOUN product, with a license term of V5R2M0, to the license repository of the iSeries server SYS1. If 1ACCOUN product is installed on the managed system, the license is also activated.

### Example 2: Adding an Activity to Send the License for Product 1CHECKS with License Term V5

```
ADDLICCRQA CRQD(MYLIB/CR3) ACTIVITY(ACT02) ACTION(*SND)
           PRDID(1CHECKS) LICTRM(V5)
           STRTIME(('23:00:00' '9/30/02'))
           NODL(NETLIB/ACCTSYS)
```

This example shows how to add an activity to send the license for product 1CHECKS with license term V5 to the systems identified in the ACCTSYS node list. Only the license keys matching the managed system serial number are added to the license repository. The license activity will be performed at 11 PM on September 30, 2002.

Top

---

## Error messages

None

Top

---

## Add Object CRQ Activity (ADDOBJCRQA)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Add Object Change Request Activity (ADDOBJCRQA) command adds an object distribution activity to a change request description. The object referred to in the activity can be an OS/400 object identified by an OS/400 object name or a global name, or a non-OS/400 object such as a PS/2 file which is identified by a global name. The object can also be an installable object which is identified by a global name. An installable object is an object created with more than one OS/400 object.

The activity can be conditioned so that it only runs after one or more other activities have completed (successfully or unsuccessfully). The activity can also be scheduled to run at a future date and time.

### Restrictions:

1. You must have \*CHANGE authority to change the request description and \*EXECUTE authority to the library.
2. If a NODL value is specified, the node can only contain entries that have a value of \*SNA for the address type.
3. An object can be specified using an OS/400 object name or a global name but not both.
4. The global name can be a maximum of 65-*n* characters in length, where *n* is the number of tokens. A maximum of 10 tokens can be specified.
5. The object to be distributed cannot reside in the QTEMP library.

### Notes:

The following notes provide information on how the command works.

1. Authorization to the product specified on the activity is not verified until the activity runs.
2. All conditions must be satisfied before the activity can run.
3. The start times indicate when the activity can be started. Actual start times can be later due to network and system delays.
4. If a global name is used, the Add Distribution Catalog Entry (ADDDSTCLGE) command can be used to indicate where the object is located or stored.
5. Only OS/400 program objects or file members such as CL and REXX can be run.
6. The save and restore history for the object is not updated when it is sent or retrieved.
7. Active message queues are not saved when libraries (\*LIB) are sent or retrieved.
8. Activities are not guaranteed to run in the same jobs. Therefore, references to objects in library QTEMP should be avoided because the scope of QTEMP is only within the job.
9. If you need to cancel an installation of a NetView Distribution Manager (NetView DM) change file or other non-OS/400 object, you must use the Add Change Management Activity (QNSADDCM) API.

Top

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACTIVITY	Activity	<i>Name, *GEN, *LAST</i>	Optional, Positional 2
ACTION	Action	<i>*SND, *RTV, *DLT, *RUN, *SNDRUN, *INS, *SNDINS, *UNINS</i>	Optional
OBJ	Object	Single values: <i>*GLOBAL, *COMPNAME</i> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Object	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
GLBNAME	Global name	<i>Element list</i>	Optional
	Element 1: Global name token 1	<i>Character value, *NETID, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 2: Global name token 2	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 3: Global name token 3	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 4: Global name token 4	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 5: Global name token 5	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 6: Global name token 6	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 7: Global name token 7	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 8: Global name token 8	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 9: Global name token 9	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 10: Global name token 10	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
OBJTYPE	Object type	<i>Character value</i>	Optional
MBR	Member	<i>Name, *FIRST, *LAST, *ALL</i>	Optional
DATATYPE	Data type	<i>*UNSPEC, *CL, *REXX</i>	Optional



Keyword	Description	Choices	Notes
COMPNAME	Component name	<i>Element list</i>	Optional
	Element 1: Component name token 1	<i>Character value, *NETID</i>	
	Element 2: Component name token 2	<i>Character value, *NETID, *CPNAME</i>	
	Element 3: Component name token 3	<i>Character value, *NETID, *CPNAME</i>	
	Element 4: Component name token 4	<i>Character value, *NETID, *CPNAME</i>	
	Element 5: Component name token 5	<i>Character value, *NETID, *CPNAME</i>	
	Element 6: Component name token 6	<i>Character value, *NETID, *CPNAME</i>	
	Element 7: Component name token 7	<i>Character value, *NETID, *CPNAME</i>	
REFLVL	Refresh level	<i>Character value, *ALL</i>	Optional
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: <b>*NONE</b> Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CPNAME	Managed system node names	Single values: <b>*NONE</b> Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Character value</i>	
TGTRLS	Target release	<b>*CURRENT</b> , *PRV, V5R1M0, V5R2M0, V5R3M0	Optional
REPLACE	Replace object	<b>*NO</b> , *YES	Optional
DTACPR	Data compression	*SNA, <b>*NONE</b>	Optional
KEEPCLGE	Keep catalog entry	*YES, <b>*NO</b>	Optional
PARM	Parameters	Values (up to 20 repetitions): <i>Not restricted</i>	Optional
RMTSTRTIME	Start time on managed system	<i>Element list</i>	Optional
	Element 1: Time zone	<b>*LCLSYS</b> , *MGDSYS	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time, *CURRENT</i>	
	Element 2: Date	<i>Date, *CURRENT, *NEXT</i>	
TEXT	Text 'description'	<i>Character value, *GEN, *BLANK</i>	Optional
COND	Activity conditions	Single values: <b>*NONE</b> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Activity	<i>Generic name, name, *PRV</i>	
	Element 2: Relational operator	<b>*EQ</b> , *GT, *LT, *LE, *NE, *GE	
	Element 3: End code	0-99, <b>*SUCCESS</b> , *FAIL, *NOTRUN, *ANY	
	Element 4: Condition mode	<b>*ALLNODES</b> , *SAMENODE	

Keyword	Description	Choices	Notes
STRTIME	Activity start time	<i>Element list</i>	Optional
	Element 1: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time, *CURRENT</i>	
	Element 2: Date	<i>Date, *CURRENT, *NEXT</i>	
	Element 2: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time, *ANY, *CURRENT</i>	
	Element 2: Date	<i>Date, *ANY, *CURRENT, *NEXT</i>	
HOLD	Hold activity	<i>*NO, *YES</i>	Optional

Top

---

## Change request description (CRQD)

Specifies the change request description object name.

The possible library values are:

**\*LIBL** All of the libraries in the user and in the system portions of the job's library list are searched.

**\*CURLIB**

The current library for the job used to locate the object.

*library-name*

Specify the name of the library to be searched.

The possible value for the change request description is:

*change-request-description*

Specify the name of the change request description object.

Top

---

## Activity (ACTIVITY)

Specifies the name of the activity to add to the change request description.

**\*GEN** An activity name is generated. The activity ID is of the form QACTxxxxxx, where xxxxxx is the first multiple of ten not already being used.

**\*LAST**

The activity is the last to run in the change request. When \*LAST is specified for the activity (ACTIVITY) parameter, the condition (COND) parameter and the start time (STRTIME) parameter cannot be specified. Only one activity named \*LAST can exist in the change request description.

*activity-name*

Specify a 10-character activity name.

Top

---

## Action (ACTION)

Specifies the object distribution function to be performed.

**\*SND** Sends the specified object to the specified managed system or systems.

- \*RTV** Retrieves the specified object from the specified managed system or systems. To retrieve an object from more than one system, a global name with an \*ANY token is required, so that each retrieved object has a unique global name. Global names with unspecified tokens (\*ALL, \*HIGHEST, or \*LOWEST) are stored in the distribution repository when they are retrieved.
- \*DLT** Deletes the specified object on the specified system or systems.
- \*RUN** Runs the specified program on the specified managed system or systems. OS/400 program objects (\*PGM), REXX programs or file members containing a CL input stream or a REXX procedure can run on an OS/400 managed system.
- \*SNDRUN**  
Sends the specified program and runs it on the specified system or systems. The program that is sent deletes on completion.
- \*INS** Installs the objects previously packaged for installation on the specified managed system or systems. Only installable objects can be installed.
- \*SNDINS**  
Sends the objects previously packaged for installation and install them on the specified managed system or systems. Only objects identified by global names can be installed.
- \*UNINS**  
Removes the objects installed on the specified managed system or systems. Only objects installed through an installable object can be uninstalled.

Top

---

## Object (OBJ)

Specifies the name of the object that runs, is sent, retrieved, or deleted. For send and retrieve actions, the object name represents the name of the object on both the central site system and the managed systems.

The possible values are:

- \*GLOBAL**  
The object is identified by the global name specified on the GLBNAME parameter.
- \*COMPNAME**  
Indicates the component name that is used to identify the installable objects that must be uninstalled from the managed system.

The possible values are:

- \*LIBL** All of the libraries in the user and in the system portions of the job's library list are searched.
- \*CURLIB**  
The current library for the job is used to locate the object.

### *library-name*

Specify the name of the library to be searched.

The possible object name value is:

### *object-name*

Specify the object name. Only characters A through Z, 0 through 9, dollar sign, at sign, and number sign can be used in the object names.

Top

---

## Global name (GLBNAME)

Specifies a global name, which is a series of tokens that uniquely identify an object in an SNA network. The global name represents the name that is used to locate the appropriate catalog entry on both the central site system and the managed systems. The catalog entry specifies the object that is to be used on that system. For example, if a retrieve action is specified, the global name is used to determine the object that is to be retrieved on the managed system. Also, the global name shows the location where it is to be stored on the central site system.

Special values in a token position indicate how to search for the object. By specifying \*ANY in a token position, the token is ignored when searching for the correct object. If multiple objects are found matching the tokens specified, an error is returned.

If an object is sent, the global name must have been previously cataloged so that it is associated with a local object name or associated with an object in the distribution repository. Retrieved objects, for which no catalog entry exists, are placed in the distribution repository. The GLBNAME parameter cannot be specified if the object name is not \*GLOBAL.

The GLBNAME parameter is not valid when the ACTION value is \*UNINS. When the OBJ is \*GLOBAL and the global name maps to an installable object, the global name must have the following structure.

ComponentName REF RefreshLevel

In this example,

- Component names are the tokens before the token with the REF value. It is used to distinguish objects of an installable object from those from another. The component can be between 1 and 7 tokens.
- The REF token is required to identify the global name as an installable object. It can only be specified from token 2 through 8 in the global name.
- The refresh level is a token with a numeric value. The refresh level shows the level of the installable object and must follow the token with the REF value.

### Token 1: Element 1

#### \*NETID

The first global name token value is a network ID generated by the command from the network attributes. The network ID is determined by the current value of the LCLNETID network attribute value.

#### **\*SERVER**

This token is stored within the change request activity with the value &SERVER, and is replaced by the short name of the change control server when the object is distributed.

#### **\*TARGET**

This token is stored within the change request activity with the value &TARGET, and is replaced by the short name of the target when the object is distributed.

#### **\*MDDATE**

This token is stored within the change request activity with the value &DATE, and is replaced when distributed by the date the object was last modified.

#### **\*MDTIME**

This token is stored within the change request activity with the value &TIME, and is replaced when distributed by the time the object was last modified.

#### *global-name-token*

Specify the first token of the global name. The first token is recommended to be the registered enterprise ID or network ID.

## Tokens 2-10: Elements 2-10

**\*ANY** Any token value matches when searching for the object where the action is to be performed. This is useful when retrieving objects for which some of the tokens in the global name are not known or vary between systems.

**\*HIGHEST**

The object with the highest token value has the action performed on it. The token must be ordered. This is useful when a token in a global name is used to indicate a different version of the object and you need to manipulate the object with the highest version level.

**\*LOWEST**

The object with the lowest token value has the action performed on it. The token must be ordered. This is useful when a token in a global name is used to indicate a different version of the object and you need to manipulate the object with the lowest version level.

**\*NETID**

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

**\*CPNAME**

The control point name of this system is used. The control point name is determined by the current value of the LCLCPNAME network attribute value.

**\*SERVER**

This token is stored within the change request activity with the value &SERVER, and is replaced by the short name of the change control server when the object is distributed.

**\*TARGET**

This token is stored within the change request activity with the value &TARGET, and is replaced by the short name of the target when the object is distributed.

**\*MDDATE**

This token is stored within the change request activity with the value &DATE, and is replaced when distributed by the date the object was last modified.

**\*MDTIME**

This token is stored within the change request activity with the value &TIME, and is replaced when distributed by the time the object was last modified.

**Note:** For more detailed information about the Change Control Server support see the Managed System Services for iSeries Use book.

*global-name-token*

Specify one of a series of 1 to 16-character tokens that uniquely identify the object on which the action is to be performed. Characters A through Z and 0 through 9 can be used. Other special values (@, #, and \$) can be used for tokens that represent network IDs and system names.

Top

---

## Object type (OBJTYPE)

Specifies the object type. It cannot be specified if a global name is used.

**\*FILEDATA**

A file member should be transferred without the file attributes. This is used to move files between an iSeries system and other non-iSeries systems. The \*FILE object type can be used with iSeries systems to preserve the file attributes.

*object-type*

Specify the OS/400 object type.

---

## Member (MBR)

Specifies the physical file member name. This cannot be specified unless the object type is \*FILE or \*FILEDATA.

**\*ALL** The action should be performed on all members within the physical file. The object type must be \*FILE. \*ALL must be used for file types that do not have members such as device files.

**Note:** \*ALL cannot be specified if the action is \*RUN or \*SNDRUN.

### \*FIRST

The action should be performed on the first member (by date added) in the physical file. The member name is determined when the activity is run. \*FIRST is not allowed when the action is \*RUN.

### \*LAST

The action should be performed on the last member (by date added) in the physical file. The member name is determined when the activity is run. \*LAST is not allowed when the action is \*RUN.

### *member-name*

Specify the member name on which the action should be performed. The member name should be specified when action is \*RUN.

Top

---

## Data type (DATATYPE)

Specifies the data type of the member. It is used to specify the type of source file that is run on the managed system. This parameter is ignored when a file is not being sent or run. In addition, this parameter cannot be specified when the object type is not \*FILEDATA.

### \*UNSPEC

Unspecified file member type. If the data type cannot be determined at the managed system, or if the name of the file where this member resides is QCLSRC, then the file member is treated as a CL batch input stream. If the source file is named QREXSRC, the file member is treated as a REXX procedure.

**\*CL** The file member contains control language, in other words a CL batch input stream.

### \*REXX

The file member contains a REXX procedure.

Top

---

## Component name (COMPNAME)

Component name, which is the set of tokens of a global name previous to the REF token. The component name is used to identify the installable objects that will be uninstalled. The component name is only valid when ACTION(\*UNINS) and OBJ(\*COMPNAME) are specified.

### Element 1: Token 1

The only special value allowed for the first token is \*NETID.

### **\*NETID**

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

### ***component-name-token***

One of a series of 1 to 16 character tokens that uniquely identifies the object on which the action is to be performed. Characters A through Z and 0 through 9 can be used. Other special values (@, #, and \$) can be used for tokens that represent network IDs and system names.

## **Elements 2-7: Tokens 2-7**

### **\*NETID**

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

### **\*CPNAME**

The control point name of this system is used. The network ID is determined by the current value of the LCLCPNAME network attribute value.

### ***component-name-token***

One of a series of 1 to 16 character tokens that uniquely identifies the object on which the action is to be performed. Characters A through Z and 0 through 9 can be used. Other special values (@, #, and \$) can be used for tokens that represent network IDs and system names.

Top

---

## **Refresh level (RFRLVL)**

The refresh level is the level of the installable object that will be uninstalled. REFLVL is only valid when ACTION(\*UNINS) and OBJ(\*COMPNAME) are specified.

**\*ALL** All the installable objects matching the component name with any refresh level will be uninstalled.

### ***refresh-level***

Specify the level of the installable object to be uninstalled. The level is a numeric value up to 16 characters.

Top

---

## **Managed systems node list (NODL)**

Specifies that the node list parameter is the object name that contains a list of systems which are the destinations for the activity. This parameter cannot be specified if the control point name (CPNAME) parameter is also specified.

The possible library values are:

### **\*NONE**

The systems on which this activity is to be performed are not specified by a node list. Individual control point names must be specified.

The possible values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

### **\*CURLIB**

The current library for the job is used to locate the node list object.

*library-name*

Specify the name of the library to be searched.

The possible node list value is:

*node-list-name*

Specify the node list object name containing the list of systems on which the activity is to be performed.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if a node list is specified.

The possible value is:

**\*NONE**

The systems on which this activity is to be performed are not identified individually. A node list must be specified.

The possible value is:

**\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

*network-identifier*

Specify the APPN network identifier of the managed system on which the activity is to be performed.

The possible control point name value is:

*control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed. For NetView Distribution Management Agents, the control point name is the change control client which supports numeric characters (0-9) in the first position of control point names that are valid in other platforms.

Top

---

## Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object. This parameter is ignored for objects with global names that are in the SystemView distribution repository or for actions other than send or retrieve.

**\*CURRENT**

The object is used on the release of the operating system currently running on your system. If V5R3M0 is running on your system, \*CURRENT means that you intend to use the object on a system with V5R3M0 installed. The object can also be used on a system with any later release of the operating system installed.

**\*PRV** The object is intended for a system which is at the previous release level compared to the local system.



**Note:** Modification levels are not supported. To specify a previous release with a modification level other than 0, such as V4R1M4, specify the value for the release rather than the special value of \*PRV.

#### *release-level*

Specify the release level in the VxRxMx format. The object is used on a system with the specified release or with any later release of the operating system installed.

Valid values depend on the current version, release, and modification level, and they change with each new release.

Top

---

## Replace object (REPLACE)

Specifies if the object should be replaced if it already exists. This parameter cannot be specified for actions other than for \*SND, \*SNDRUN, and \*RTV.

**\*NO** An error is returned if the object already exists.

**\*YES** The object is replaced if it already exists.

Top

---

## Data compression (DTACPR)

Specifies if data is compressed when sending or receiving. This parameter cannot be specified for actions other than \*SND, \*SNDRUN, and \*RTV. SNA compression with a prime compression character of blank is performed.

#### **\*NONE**

The file data is not compressed when sent or when retrieved.

**\*SNA** The file data is compressed when sent or when retrieved.

#### **Note:**

Objects that are globally named can have compression information specified when they were added to the distribution repository using the Add Distribution Catalog Entry (ADDDSTCLGE) command.

Top

---

## Keep catalog entry (KEEPCLGE)

Specifies if the catalog entry and associated save file corresponding to the installable object will be kept in the specified system or systems. The KEEPCLGE parameter is only valid when ACTION(\*SNDINS) or ACTION(\*INS) is specified.

**\*NO** The catalog entry and associated save file are not kept.

**\*YES** The catalog entry and associated save file are kept.

Top

---

## Parameters (PARM)

Specifies that the parameters to be passed when starting the program. This cannot be specified if the action is not \*RUN or \*SNDRUN. Up to 20 parameters can be specified.

*parameter*

Specify a 1 to 253 character parameter. The prompt panel initially allows 50 characters to be entered. By entering an ampersand (&) in position 1, the field expands for larger parameters.

Top

---

## Start time on managed system (RMTSTRTIME)

Specifies the date and time when the activity can begin running on the managed system. The current date and time values and next date values are determined when the activity begins running at the central site system based on the central site date and time.

The possible values of the time zone of the remote start time are:

**\*LCLSYS**

The remote start time is specified in the time zone of the central site system.

**\*MGDSYS**

The remote start time is specified in the time zone of the managed system.

The possible start after time values are:

This is the definition of the time after which the activity is to start.

**\*CURRENT**

This function can start on the managed system at any time on or after the time this activity is started on the central site system on the date specified in element 2.

*start-after-time*

Specify the time when this function can start on the managed system. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits such as hh:mm:ss.

The possible start after date values are:

**\*CURRENT**

This function starts on the managed system on any date on or after the activity starts on the central site system.

**\*NEXT**

This function starts on the managed system on any date after the activity starts on the central site.

*start-after-date*

Specify the date after the functions start on the managed system. The date must be specified in the job date format.

**Notes:**

1. The special values \*CURRENT and \*NEXT are specified for the date and the time if the time zone value \*MGDSYS is specified.
2. This parameter can only be specified if \*RUN, \*SNDRUN, \*INS, \*SNDINS, and \*UNINS actions are specified.

Top

---

## Text 'description' (TEXT)

Specifies the activity description.

**\*GEN** A description is generated based on the action specified.

### *text-description*

Specify a 50-character description of the activity.

Top

---

## Activity conditions (COND)

Specifies which conditions must be met before this activity can be performed. Each condition identifies an activity which must run before this activity, and the value the end code from that activity must have to allow this activity to run. The default condition is that the previous activity (in alphabetical order) must complete successfully before this activity can be run.

The possible single value is:

### **\*NONE**

There are no conditions for this activity.

The possible conditioning activity values are:

These are the activities which must be run before this activity.

**\*PRV** This activity is conditioned on the previous activity. Activities are ordered alphabetically by activity name. If the activity being added is the first activity, a previous activity does not exist and any condition with \*PRV is marked as having been met.

### *conditioning-activity-name*

Specify the name of the activity which must run before this activity. The activity name specified in the activity (ACTIVITY) parameter cannot be specified in the conditioning activity name. An activity cannot be conditioned on itself.

### *generic\*-activity-name*

Specify the generic name of the activities which must run before this activity.

The possible relational operator values are:

This element is the relational operator to use when comparing the end code from the conditioning activity.

**\*EQ** Equal

**\*GT** Greater than

**\*LT** Less than

**\*NE** Not equal

**\*GE** Greater than or equal

**\*LE** Less than or equal

The possible condition code values are:

This element is the value compared to the actual end code of the conditioning activity.

### **\*SUCCESS**

The activity ended successfully (0 <= end code <= 9). This end code can only be specified with relational operator \*EQ or \*NE.

**\*FAIL** The activity failed (10 <= end code <= 89). This end code can only be specified with relational operator \*EQ or \*NE.

### **\*NOTRUN**

The activity is never started (90 <= end code <= 99). This end code is only specified with relational operator \*EQ or \*NE.

**\*ANY** The activity ended with any end code. This end code is only specified with relational operator \*EQ.

### ***end-code***

Specify an integer value (0-99) that indicates the result of an activity (success or failure). The end code ranges and descriptions are:

- 00** Activity completed successfully.
- 01-09** Activity completed with warning messages.
- 10-29** Activity did not complete successfully.
- 30-39** Activity was canceled by a user before it completed.
  - 30 = Activity ended with \*CNTRLD option
  - 35 = Activity ended with \*IMMED option
  - 39 = Activity ended with \*FRCFAIL option
- 40-49** Activity was not run due to errors detected by the application.
  - 40 = Activity not run for security reasons
- 90-99** Activity was not run because conditions or schedules were not met.
  - 95 = Scheduled start time expired
  - 99 = Conditions cannot be met

The possible condition mode values are:

This element indicates which systems the conditioning activity must have completed on before this activity can be performed.

### **\*ALLNODES**

The conditioning activity specified must complete on all nodes before this activity runs.

### **\*SAMENODE**

When the conditioning activity specified completes for a given node, the activity specified on the ACTIVITY parameter may run for that same node even though the conditioning activity specified may not have completed for all other nodes. In the case where this activity lists a node not in the conditioning activity, this activity may run for that node; the condition is ignored.

Top

---

## **Activity start time (STRTIME)**

Specifies the date and time when this activity can be started on the central site system. The current date and time values, and next date values are determined when the change request is submitted.

The possible start after time values are:

**\*CURRENT**

This activity can start any time on or after the time when the change request is submitted.

***start-after-time***

Specify the time when this activity can start. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits such as hh:mm:ss.

The possible start after date values are:

**\*CURRENT**

This activity can start on or after the date on which the change request is submitted.

**\*NEXT**

The activity can start on any date after the date when the change request is submitted.

***start-after-date***

Specify the date after this activity can start. The date must be specified in the job date format.

The possible start before time values are:

This element is ignored if the start before date is \*ANY.

**\*ANY** The activity can start at any time on or before the start before date.

**\*CURRENT**

The activity must start before the time when the change request was submitted on the date specified on the start before data element.

***start-before-time***

Specify the time before the activity must start. If the activity cannot be started before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start before date values are:

**\*ANY** The activity can start at any time after the start after time and the start after date.

**\*CURRENT**

The activity must start on the date the change request is submitted.

**\*NEXT**

The activity must start by the day after the date the change request is submitted.

***start-before-date***

Specify the date before the activity must start. If the activity cannot start by this date, it never starts. The date must be specified in the job date format.

Top

---

## Hold activity (HOLD)

Specifies that the activity be held when the change request is submitted.

**\*NO** The activity is not held. It runs when all conditions and the start time are met.

**\*YES** The activity is held for all nodes when the change request is submitted. It must be released by you before it runs.

---

## Examples

### Example 1: Adding an Activity to a Change Request Description

```
ADDOBJCRQA  CRQD(MYLIB/CR1)  ACTIVITY(ACT01)  ACTION(*RTV)
             OBJ(QGPL/QXYZ)  OBJTYPE(*JOB)
             CPNAME((*NETATR SYS1))
```

This example shows how to add an activity to change request description CR1 in library MYLIB which retrieves the QGPL/QXYZ job description from the iSeries server SYS1 in the local network.

### Example 2: Adding an Activity to an Accounting System

```
ADDOBJCRQA  CRQD(MYLIB/CR2)  ACTIVITY(ACT02)  ACTION(*SND)
             OBJ(ACCTLIB/TAXFILE)  OBJTYPE(*FILE)
             MBR(DEDUCTIONS)  TGTRLS(*PRV)
             STRTIME((23:00:00 9/30/02))
             NODL(NETLIB/ACCTSYS)
```

This example shows how to add an activity to send a tax table to all of the iSeries accounting servers in the ACCTSYS node list at 11 p.m. on 30 September 2002. The accounting systems are at the previous release level.

### Example 3: Adding an Activity to Retrieve a Program

```
ADDOBJCRQA  CRQD(MYLIB/CR3)  ACTIVITY(*GEN)  ACTION(*RTV)
             OBJ(*GLOBAL)
             GLBNAME(CUSTNET PCSOFT WDWAPP VER5 020314)
             CPNAME((CUSTNET DEVPS2))

ADDOBJCRQA  CRQD(MYLIB/CR3)  ACTIVITY(*GEN)  ACTION(*SND)
             OBJ(*GLOBAL)
             GLBNAME(CUSTNET PCSOFT WDWAPP VER5 020314)
             NODL(NETLIB/PS2SE)

ADDOBJCRQA  CRQD(MYLIB/CR3)  ACTIVITY(*GEN)  ACTION(*RUN)
             OBJ(*GLOBAL)
             GLBNAME(CUSTNET PCSOFT WDWAPP VER5 020314)
             COND((*PRV *EQ *SUCCESS *SAMENODE))
             RMTSTRTIME((*MGDSYS (23:00:00 10/20/02))
             NODL(NETLIB/PS2SE)
```

This command adds activities to retrieve a program from a PS/2, and sends it to all of the PS/2s in the southeast area. The PS/2s are identified in the PS2SE node list. It runs it on the PS/2s at 11 p.m. in the time zone where the PS/2 is located only if the send to the PS/2 was successful. The activity names are generated.

### Example 4: Adding an Activity to Retrieve a File from Multiple Systems

```
ADDOBJCRQA  CRQD(MYLIB/CR4)  ACTIVITY(ONLY)  ACTION(*RTV)
             OBJ(*GLOBAL)  GLBNAME(CUSTNET SALES *ANY *HIGHEST)
             STRTIME((22:00:00 *CURRENT) (06:00:00 *NEXT))
             NODL(NETLIB/STORES)
```

This command adds an activity to retrieve the most recent nightly sales file from each system identified in the STORES node list. The files are cataloged as CUSTNET SALES system-name date-created. The file must be retrieved after 10 p.m. on the day the request is submitted but before 6 a.m. the next morning when the stores open.

### Example 5: Adding an Activity to Send an Installable Object

```
ADDOBJCRQA  CRQD(MYLIB/CR1)  ACTIVITY(01)  ACTION(*SND)
             OBJ(*GLOBAL)
             GLBNAME(CUSTNET ASOBJ PCKOBJ 01269 REF 0001)
             CPNAME((*NETATR SYS1))
```

This command adds an activity to send an object that was previously packaged for installation in the central site to the iSeries server SYS1.

#### **Example 6: Adding an Activity to Send and Install an Installable Object**

```
ADDOBJCRQA  CRQD(MYLIB/CR2)  ACTIVITY(ACT02)  ACTION(*SNDINS)
             OBJ(*GLOBAL)
             GLBNAME(CUSTNET ASOBJ PCKOBJ 01270 REF 0002)
             NODL(NETLIB/ACCTSYS)  KEEPCLGE(*NO)
```

This command adds an activity to send and install an installable object that was previously packaged for installation to the node list ACCTSYS. The catalog entry in the managed system pointing to the installable object is not kept.

#### **Example 7: Uninstalling an Installable Object**

```
ADDOBJCRQA  CRQD(MYLIB/CR4)  ACTIVITY(ACT04)  ACTION(*UNINS)
             OBJ(*COMPNAME)
             COMPNAME(CUSTNET ASOBJ PCKOBJ 03000)
             REFLVL(*ALL)  CPNAME((*NETATR SYS4))
```

This command adds an activity to uninstall all the installable objects from the system SYS4 whose global name matches the component name CUSTNET ASOBJ PCKOBJ 03000, no matter what release level it has.

[Top](#)

---

## **Error messages**

None

[Top](#)





---

## Add Product CRQ Activity (ADDPRDCRQA)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Add Product Change Request Activity (ADDPRDCRQA) command adds an activity to a change request description which performs a product distribution function. The activity can be conditioned so that it runs after one or more other activities have completed (successfully or unsuccessfully). The activity can also be scheduled to run at a future date and time.

### Restrictions:

1. This command is shipped with public \*EXCLUDE authority.
2. You must have \*CHANGE authority to the change request description object and \*EXECUTE authority to the library.
3. The product must have been previously packaged for distribution. You can use the Package Product for Distribution (PKGPRDDST) command to package the product.
4. If a NODL value is specified, the node list can only contain entries that have a value of \*SNA for the address type.

### Notes:

The following notes provide information on how the command works.

1. Authorization to the product specified on the activity is not verified until the activity runs.
2. All conditions must be satisfied before the activity can be performed.
3. The start times indicate when the activity can be started. Actual start times can be later due to network and system delays.
4. Action \*DLTCLGE only deletes the save file containing the licensed program and deletes the distribution catalog entry that maps this save file. It does not delete the product.
5. If TGTRLS(\*ONLY) is specified, the action parameter must be a value other than \*INS.
6. When it is requested to send the license key of the product, a record with the license key information must exist in the central site license repository by the time this activity runs.
7. All the existing records in the central site license repository containing the license key information for the specified product is sent to the specified managed system or systems, but only those that match the system serial number are added to the managed system license repository.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACTIVITY	Activity	<i>Name, *GEN, *LAST</i>	Optional, Positional 2
ACTION	Action	*SND, *RTV, *DLTCLGE, *INS, *SNDINS	Optional, Positional 3

Keyword	Description	Choices	Notes
PRDID	Product ID	Character value	Optional, Positional 4
RLS	Release level	Character value, <u>*ONLY</u>	Optional
OPTION	Product option	1-99, <u>*BASE</u>	Optional
LODTYPE	Product load type	<u>*ALL</u> , *CODE, *LNG	Optional
LODID	Load ID	2900-2999, <u>*ALL</u> , *CODE	Optional
TGTRLS	Target release	<u>*CURRENT</u> , *PRV, V5R1M0, V5R2M0, V5R3M0, *ONLY	Optional
NODL	Managed systems node list	Element list	Optional
	Element 1: Node list	Single values: <u>*NONE</u> Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
CPNAME	Managed system node names	Single values: <u>*NONE</u> Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	Communications name, <u>*NETATR</u>	
	Element 2: Control point	Communications name	
KEEPCLGE	Keep catalog entry	*YES, <u>*NO</u>	Optional
SNDLICENSE	Send license key	*YES, <u>*NO</u>	Optional
RMTSTRTIME	Start time on managed system	Element list	Optional
	Element 1: Time zone	<u>*LCLSYS</u> , *MGDSYS	
	Element 2: Start after	Element list	
	Element 1: Time	Time, <u>*CURRENT</u>	
	Element 2: Date	Date, <u>*CURRENT</u> , *NEXT	
TEXT	Text 'description'	Character value, <u>*GEN</u> , *BLANK	Optional
COND	Activity conditions	Single values: <u>*NONE</u> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Activity	Generic name, name, <u>*PRV</u>	
	Element 2: Relational operator	<u>*EQ</u> , *GT, *LT, *LE, *NE, *GE	
	Element 3: End code	0-99, <u>*SUCCESS</u> , *FAIL, *NOTRUN, *ANY	
	Element 4: Condition mode	<u>*ALLNODES</u> , *SAMENODE	
STRTIME	Activity start time	Element list	Optional
	Element 1: Start after	Element list	
	Element 1: Time	Time, <u>*CURRENT</u>	
	Element 2: Date	Date, <u>*CURRENT</u> , *NEXT	
	Element 2: Start before	Element list	
	Element 1: Time	Time, <u>*ANY</u> , *CURRENT	
	Element 2: Date	Date, <u>*ANY</u> , *CURRENT, *NEXT	
HOLD	Hold activity	<u>*NO</u> , *YES	Optional

Top

---

## Change request description (CRQD)

Specifies the change request description object name.

The possible library values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched.

**\*CURLIB**

The current library for the job is used to locate the object.

*library-name*

Specify that only the library named in this parameter is searched.

*change-request-description*

Specify the name of the change request description object.

Top

---

## Activity (ACTIVITY)

Specifies the name of the activity to add to the change request description.

**\*GEN** An activity name is generated. The activity name is of the form QACTxxxxxx where xxxxxx is the first multiple of ten not already being used.

**\*LAST**

The activity is the last to run in the change request. When \*LAST is specified for the activity (ACTIVITY) parameter, the condition (COND) parameter and the start time (STRTIME) parameter cannot be specified. Only one activity named \*LAST can exist in the change request description.

*activity-name*

Specify a 10-character activity name.

Top

---

## Action (ACTION)

Specifies the product distribution function that is to be performed.

**\*SND** Send the product to the specified managed systems.

**\*RTV** Retrieve the product from the specified managed system.

**\*DLTCLGE**

Delete the catalog entry and its associated save file which contains the product packaged for distribution from the specified managed systems.

**\*INS** Install the product on the specified managed systems.

**\*SNDINS**

Send and install the specified product on the specified managed systems.

Top

---

## Product ID (PRDID)

Specifies the 7-character identifier of the product for which the action is performed.

The possible product ID value is:

### *product-ID*

Specify the 7-character product ID that is used in the activity.

Top

---

## Release level (RLS)

Specifies which version, release, and modification level of the product is used.

The possible values are:

### \*ONLY

The release level of the product that is installed on your system.

### *release-level*

Specify the release level in the format VxRxMy, where Vx is the version number, Rx is the release number, and My is the modification number. Valid values for x are the numbers 0 through 9. Valid values for y range from the numbers 0 through 9 and the letters A through Z.

Top

---

## Product option (OPTION)

Specifies which of the optional parts of the product given in the PRDID parameter are used.

The possible values are:

### \*BASE

Only the base part of the product is used.

### *product-option-number*

Specify the option number for the product load being used. Valid values range from 1 through 99.

Top

---

## Product load type (LODTYPE)

Specifies the product load objects being used.

The possible values are:

\*ALL Code and language objects specified on the LODID parameter are used.

### \*CODE

The program objects associated with this product load are used.

\*LNG The objects associated with the national language version (NLV) identified on the LODID parameter are used.

Top

---

## Load ID (LODID)

Specifies the load identifier used.

The possible values are:

**\*ALL** All languages for this product option are saved.

**\*CODE**

The code load is used.

***product-load-ID***

Specify the load ID of the product when LODTYPE(\*LNG) or LODTYPE(\*ALL) is used. The load ID must be one of the valid IBM national language versions and be specified in the form 29xx. The value of x can be 0 through 9.

Top

---

## Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the product.

**\*ONLY**

The release is determined by the release of the existing product. This value is not valid if more than one release exists for the same product.

**\*CURRENT**

The product is to be used on the release of the operating system currently running on your system.

**\*PRV** The product is to be used on the previous release with modification level 0 of the operating system.

***release-level***

Specify the release level in the format VxRxMx. The product can be used on a system with the specified release or with any later release of the operating system installed. Valid values depend on the current version, release, and modification level, and can change with each new release.

Top

---

## Managed systems node list (NODL)

Specifies that the node list parameter is the object name that contains a list of systems which are the destinations for the activity. This parameter cannot be specified if the control point (CPNAME) parameter or ACTION(\*RTV) is specified.

The possible values are:

**\*NONE**

The systems on which this activity is to be performed are not specified by a node list. Individual control point names must be specified.

The possible library values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

**\*CURLIB**

The current library for the job is used to locate the node list object.

***library-name***

Specify the name of the library to be searched.

***node-list-name***

Specify the node list object name containing the list of systems on which the command is to be performed.

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if a node list is specified. If ACTION(\*RTV) is used only one control point name can be specified.

### \*NONE

The systems on which this activity is to be performed are not identified individually. A node list must be specified.

The possible network identifier value is:

### \*NETATR

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

### *network-identifier*

Specify the APPN network identifier of the managed system on which the activity is to be performed.

The possible control point name value is:

### *control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed.

Top

---

## Keep catalog entry (KEEPCLGE)

Specifies if the distribution catalog entry and its associated save file corresponding to the product is kept on the specified systems. This is only valid if ACTION(\*INS) or ACTION(\*SNDINS) is specified.

The possible values are:

\*NO The catalog entry and associated save file are not kept.

\*YES The catalog entry and associated save file are kept.

Top

---

## Send license key (SNDLICKEY)

Specifies if the license key is to be sent with the product.

\*YES The license key is sent with the product.

\*NO The license key is not sent with the product.

Top

---

## Start time on managed system (RMTSTRTIME)

Specifies the date and time after which the activity can begin running on the managed system. The current date and time values and next date values are determined when the activity begins running on the central site system and are based on the central site date and time.

The possible time zone values are:

### \*LCLSYS

The remote start time is specified in the time zone of the central site system.

### \*MGDSYS

The remote start time is specified in the time zone of the managed system.

The possible start after time values are:

### \*CURRENT

This function can start on the managed system at any time on or after the time this activity is started on the central site system on the date specified in element 2.

### **start-after-time**

Specify the time after which this function can start on the managed system. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

### \*CURRENT

This function starts on the managed system on any date on or after the activity starts on the central site system.

### \*NEXT

This function starts on the managed system on any date after the activity starts on the central site.

### **start-after-date**

Specify the date after the functions may start on the managed system. The date must be specified in the job date format.

### **Notes:**

1. The special values \*CURRENT and \*NEXT are specified for the date and time if the time zone value \*MGDSYS is specified.
2. This parameter can only be specified if \*INS or \*SNDINS actions are specified.

Top

---

## Text 'description' (TEXT)

Specifies an activity description.

\*GEN A text description is generated based on the action chosen.

### **text-description**

Specify a 50 character description of the activity.

Top

---

## Activity conditions (COND)

Specifies which conditions must be met before this activity can be performed. Each condition identifies an activity which must run before this activity and the value the end code from that activity must have to allow this activity to run. The default condition is that the previous activity (in alphabetical order) must complete successfully before this activity can be run.

The possible single value is:

### \*NONE

There are no conditions for this activity.

The possible conditioning activity values are:

\*PRV This activity is conditioned on the previous activity. Activities are ordered alphabetically by activity name. If the activity being added is the first activity, a previous activity does not exist and any condition with \*PRV is marked as having been met.

### *conditioning-activity-name*

Specify the name of the activity which must run before this activity.

### *generic\*-activity-name*

Specify the generic name of the activities which must run before this activity.

The possible relational operator values are:

This element is the relational operator to use when comparing the end code from the conditioning activity.

\*EQ Equal

\*GT Greater than

\*LT Less than

\*NE Not equal

\*GE Greater than or equal

\*LE Less than or equal

The possible condition code values are:

This element is the value compared to the actual end code of the conditioning activity.

### \*SUCCESS

The activity ended successfully (0 <= end code <= 9). This end code can only be specified with relational operator \*EQ or \*NE.

\*FAIL The activity failed (10 <= end code <= 89). This end code can only be specified with relational operator \*EQ or \*NE.

### \*NOTRUN

The activity never started (90 <= end code <= 99). This end code can only be specified with relational operator \*EQ or \*NE.

\*ANY The activity ended with any end code. This end code can only be specified with relational operator \*EQ.

### *end-code*

Specify an integer value (0-99) that indicates the result of an activity (success or failure). The end code ranges and descriptions are:

00 Activity completed successfully.



- 01-09 Activity completed with warning messages.
- 10-29 Activity did not complete successfully.
- 30-39 Activity was canceled by a user before it completed.
  - 30 = Activity ended with \*CNTRLD option
  - 35 = Activity ended with \*IMMED option
  - 39 = Activity ended with \*FRCFAIL option
- 40-49 Activity was not run due to errors detected by the application.
  - 40 = Activity not run for security reasons
- 90-99 Activity was not run because conditions or schedules were not met.
  - 95 = Scheduled start time expired
  - 99 = Conditions cannot be met

The possible condition mode values are:

This element indicates which systems the conditioning activity must have completed on before this activity can be performed.

**\*ALLNODES**

The conditioning activity specified must complete on all nodes before this activity runs.

**\*SAMENODE**

When the conditioning activity specified completes for a given node, the activity specified on the ACTIVITY parameter may run for that same node even though the conditioning activity specified may not have completed for all other nodes. In the case where this activity lists a node not in the conditioning activity, this activity may run for that node; the condition is ignored.

Top

## Activity start time (STRTIME)

Specifies the date and time when this activity can be started on the central site system. The current date and time values and next date values are determined when the change request is submitted.

The possible start after time values are:

**\*CURRENT**

This activity can start any time on or after the time when the change request is submitted.

***start-after-time***

Specify the time when this activity can start. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

**\*CURRENT**

This activity can start on or after the date on which the change request is submitted.

**\*NEXT**

The activity can start on any date after the date the change request is submitted.

***start-after-date***

Specify the date after this activity can start. The date must be specified in the job date format.

The possible start before time values are:

This element is ignored if the start before date is \*ANY.

**\*ANY** The activity can start at any time on or before the start before date.

**\*CURRENT**

The activity must start before the time at which the change request was submitted on the date specified on the start before date element. This value cannot be specified if the start before date is \*CURRENT.

***start-before-time***

Specify the time before the activity must start. If the activity cannot be started before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. The time can be specified with or without a time separator such as a colon (:). Seconds are optional. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start before date values are:

**\*ANY** The activity can start at any time after the start after time and the start after date.

**\*CURRENT**

The activity must start on the date the change request is submitted.

**\*NEXT**

The activity must start by the day after the date the change request is submitted.

***start-before-date***

Specify the date before the activity must start. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

Top

---

## Hold activity (HOLD)

Specifies that the activity be held when the change request is submitted.

**\*NO** The activity is not held. It runs when all conditions and the start time are met.

**\*YES** The activity is held for all nodes when the change request is submitted. It must be released by you before it runs.

Top

---

## Examples

### Example 1: Adding an Activity to Retrieve

```
ADDPRDCRQA  CRQD(MYLIB/CR1)  ACTIVITY(ACT01)  ACTION(*RTV)
              PRDID(1ACCOUN)  RLS(V5R2M0)  OPTION(*BASE)
              LODTYPE(*ALL)  LODID(*ALL)
              CPNAME((*NETATR SYS1))
```

Add an activity to retrieve the base option for the 1ACCOUN product, release V5R2M0, for both the code and the language parts from the iSeries server SYS1.

### Example 2: Adding an Activity to Send and Install

```
ADDPRDCRQA  CRQD(MYLIB/CR2)  ACTIVITY(ACT01)  ACTION(*SNDINS)
             PRDID(1CHECKS)  RLS(V5R2M0)  OPTION(0010)
             LODTYPE(*ALL)  LODID(*ALL)
             CPNAME((*NETATR SYS2))  SNDLICKEY(*YES)
```

This command adds an activity to send and install option 10 of the 1CHECKS product, release V5R2M0, for both the code and the language parts from the iSeries server SYS2. The license key for this product is also sent and is added to the managed system license repository.

### Example 3: Adding an Activity to Install Language Objects

```
ADDPRDCRQA  CRQD(MYLIB/CR3)  ACTIVITY(ACT02)  ACTION(*SND)
             PRDID(1ACCOUN)  RLS(V5R2M0)  OPTION(*BASE)
             LODTYPE(*ALL)  LODID(2924)
             NODL(NETLIB/ACCTSYS)  STRTIME((23:00:00 9/30/02))
```

This command adds an activity to send the language objects for the English version and the code objects of the base option of the 1ACCOUN product, to all the systems in the node list (NODL) ACCTSYS network at 11 p.m. on 30 September 2002. The license key for this product is also sent and is added to the managed system license repository.

[Top](#)

---

## Error messages

None

[Top](#)



---

## Add License Information (ADDPRDLICI)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Add Product License Information (ADDPRDLICI) command adds license information to the product or feature. This command can be used before or after the product or options belonging to a feature are packaged using the Package Product Option (PKGPRDOPT) command, but before the product or options belonging to a feature are installed. If the product has keyed (\*KEYED) compliance, a product license information handle is included in the completion message received from this command.

For more information about the Add Product License Information (QLZADDLI) API is also available in the API topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

**Restriction:** This command is shipped with public \*EXCLUDE authority.

Top

---

### Parameters

Keyword	Description	Choices	Notes
PRDID	Product ID	<i>Character value</i>	Required, Positional 1
RLS	Release level	<i>Character value</i>	Required, Positional 2
FEATURE	Feature	<i>Character value, 5001</i>	Optional
USGTYPE	Usage type	<i>*CONCURRENT, *REGISTERED</i>	Optional
COMPLIANCE	Compliance type	<i>*OPRACTION, *WARNING, *KEYED</i>	Optional
DFTUSGLMT	Default usage limit	<i>0-999999, 1, *NOMAX</i>	Optional
LICTRM	License term	<i>*VERSION, *RELEASE, *MOD</i>	Optional
MSGID	Feature message ID	<i>Name, *BASEOPT</i>	Optional
ALWLICRLS	Allow license release	<i>*NO, *YES</i>	Optional
VNDPWD	Vendor password	<i>Simple name, *NONE</i>	Optional
GRACE	Grace period	<i>Element list</i>	Optional
	Element 1: Number of days	<i>0-999, 0</i>	
	Element 2: Allow for default usage limit	<i>*NO, *YES</i>	

Top

---

### Product ID (PRDID)

Specifies the product identifier of the product loads to which the license information is being added.

**Note:** The identifier must be exactly 7 characters long.

Top

---

## Release level (RLS)

Specifies the version, the release, and the modification level of the product or feature to which the license information is being added.

The possible values are:

### *version-release-modification*

Specify the version, release, and modification level of the product or feature where the license information is being added. The release level must be in the format VxRyMz, where valid values for x and y range from 0 through 9 and valid values for z range from 0 through 9 and A through Z

Top

---

## Feature (FEATURE)

Specifies the feature for which the license information is being added. The license information is added to all product loads with the specified product ID and feature.

**5001** License information for the 5001 feature of the product specified on the PRDID parameter is added.

### *load-ID*

Specify that the license information for the feature on the PRDID parameter associated with this number is to be added. Valid values range from 5001 through 9999.

Top

---

## Usage type (USGTYPE)

Specifies the type of license usage.

The possible values are:

### **\*CONCURRENT**

The usage limit is for the number of unique jobs accessing the product or feature at one time.

### **\*REGISTERED**

The usage limit is for the number of unique license users registered by the product or feature.

Top

---

## Compliance type (COMPLIANCE)

Specifies the action that must be taken when the usage limit is exceeded.

The possible values are:

### **\*OPRACTION**

The usage limit cannot be exceeded. If you attempt to use the product or feature after the usage limit has been reached, you are prevented from accessing the product or feature until the appropriate action is taken by the system administrator to increase the usage limit.

### **\*WARNING**

If you attempt to use the product or feature after the usage limit has been reached, you are allowed access. A warning message indicating that the usage limit has been exceeded is sent to the licensing information owner for notification.

### **\*KEYED**

A license key is required from the software provider to install or to increase the usage limit of the product or feature. The software provided can permit a grace period which allows the usage limit of a product or feature to be exceeded for n days before a new license key is required to allow additional users access to the product. If a new key with an increased usage limit is not installed prior to the end of the grace period, the usage limit of the product reverts back to the entitled usage limit.

Top

---

## **Default usage limit (DFTUSGLMT)**

Specifies the usage limit that is in effect the first time the product or feature is installed for a license term.

**Note:** For \*KEYED compliance, this is the number of users that are able to use the product or feature before the license key is installed using the Add License Key (ADDLICENSE) command or Add License Key Information (QLZAADDK) API. To prevent any access to the product or feature without a key, the default usage limit should be set to 0.

The possible values are:

1 One user is allowed to access the product or feature.

### **\*NOMAX**

Any number of users are allowed to access the product or feature.

### *default-usage-limit*

Specify the default usage limit number. The valid values range from 0 through 999&finespace.999.

Top

---

## **License term (LICTRM)**

Specifies which levels of the product or feature share license information.

The possible values are:

### **\*VERSION**

All VxRx levels of the product, where Vx is the version level and Rx is the release level, specified on the RLS parameter whose license term is also \*VERSION share the same license information. When the version level of the product or feature changes, customers installing the new release must use the Change License Information (CHGLICINF) command for a compliance type of \*WARNING or \*OPRACION, or the Add License Key (ADDLICENSE) command for a compliance type of \*KEYED, to set the license information.

### **\*RELEASE**

All VxRx levels of the product, where Vx is the version level and Rx is the release level, specified on the RLS parameter whose license term is also \*RELEASE share the same license information. When the version level or the release level of the product or feature changes, customers installing the new release must use the Change License Information (CHGLICINF) command for a compliance type of \*WARNING or \*OPRACION or the Add License Key (ADDLICENSE) command for a compliance type of \*KEYED to set the license information.

### **\*MOD**

Each level of the product or feature is licensed. No license information is shared between levels. Customers installing a new level of the product or feature must use the Change License

Information (CHGLICINF) command for compliance type of \*WARNING or \*OPRACTION, or the Add License Key (ADDLICKEY) command for compliance type of \*KEYED to set the license information.

Top

---

## Feature message ID (MSGID)

Specifies the message identifier of the message that describes the product feature. The message file for this message ID is specified in the product definition.

The possible values are:

### \*BASEOPT

There is no message describing this product feature. The message used for describing the base option of this product is used instead.

### *feature-message-id*

Specify the message identifier of the message that describes this product feature.

Top

---

## Allow license release (ALWLICRLS)

Specifies whether a license that is previously requested can be released using the WRKLICINF command. When the license is released, the usage count is decreased. This value can only be specified for a registered license type.

If it is necessary for the product to be notified that a license has been released, \*NO should be specified and the product must provide its own mechanism for releasing a license.

The possible values are:

\*NO A license cannot be released using the WRKLICINF command.

\*YES A license can be released using the WRKLICINF command.

Top

---

## Vendor password (VNDPWD)

Specifies the software vendor's password. This password is encrypted and stored with the product. It is used in validating Add License Key requests and must be the same password used to generate keys for this product and feature on the Generate License Key (GENLICKEY) command.

The possible values are:

### \*NONE

There is no vendor password needed for the Add License Key Information (ADDLICKEY) command.

### *vendor-password*

Specify the vendor password. It must begin with an alphabetic character such as A through Z, \$, # or @. It is then followed by no more than 9 alphanumeric characters such as A through Z, 0 through 9, \$, # @ or \_.

Top



---

## Grace period (GRACE)

Specifies the number of days after a product first exceeds its usage limit that you have to obtain a new license key. If a new license key is not obtained from the software vendor by the time the grace period is expired, no users over the usage limit are allowed to access the product or feature. When the usage limit is first exceeded, the date the grace period expires is calculated by adding the number of days in the grace period to today's date.

The possible number grace period days values are:

**0** There is no grace period. The user is not allowed to exceed the usage limit.

### *grace-period*

Specify the number of days in the grace period. Valid values range from 0 through 999.

The possible values for allowing for a default usage limit are:

**\*NO** The grace period is not allowed for use with the default usage limit. This means that you are not allowed to exceed the default usage limit before the license key is installed using the Add License Key command (ADDLICENSE) or Add License Key Information (QLZAADDK) API.

**\*YES** The grace period is allowed for use with the default usage limit. This means that you can exceed the default usage limit for the number of days in the grace period before the license key is installed.

Top

---

## Examples

### Example 1: Adding Product License Information

```
ADDPRDLICI PRDID(1MYPROD) RLS(V5R2M0)
```

This command adds product license information to the product with product identifier 1MYPROD for release V5R2M0.

### Example 2: Adding Product License Information for the V5R2M0 Release

```
ADDPRDLICI PRDID(2MYPROD) RLS(V5R2M0) COMPLIANCE(*KEYED)  
VNDOWD(PRODUCTPWD) GRACE(30 *YES)
```

This command adds product license information to the product with the product identifier 1MYPROD for release V5R2M0. This product supports keyed compliance. It can exceed its usage limit and default usage for 30 days.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF9E1A

License information conflict found.

Top



---

## Add PTF CRQ Activity (ADDPTFCRQA)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Add Program Temporary Fix Change Request Activity (ADDPTFCRQA) command adds an activity to a change request description which performs a PTF distribution function.

### Restrictions:

1. You must have \*CHANGE authority to the change request description and \*EXECUTE authority to the library.
2. The PTF must be for an iSeries product package using the System Manager licensed program. The PTF must be supported using the Work with Supported Product (WRKSPTPRD) command. Software redesigns can be managed for other types of systems by using globally named objects on the Add Object Change Request Activity (ADDOBJCRQA) command or by using Add Change Management Activity (QNSADDCM) API.
3. A PTF save file must exist and be released in order to be sent to another system.
4. A PTF can only be retrieved from a single managed system.
5. If a NODL value is specified, the node list can only contain entries that have a value of \*SNA for the address type.
6. If the destination node does not have IBM SystemView Managed System Services installed, only the send action is available. The activity is considered successful when the PTF is sent, not when it arrives.
7. If you are distributing a PTF and the distribution queue is set to \*SNADS in the Work with Service Requesters (WRKSRVRQS) command, you must have \*USE authority to the Send PTF (SNDPTF) command.
8. If you are distributing a PTF and the distribution queue is set to \*SVDS in the Work with Service Requesters (WRKSRVRQS) command, you must have \*USE authority to the Copy PTF (CPYPTF) command.

### Notes:

The following notes provide information on how the command works.

1. Authorization to the product specified on the activity is not verified until the activity runs.
2. All conditions must be met before the activity runs.
3. The start times indicate when the activity can be started. Actual start times can be later due to network and system delays.
4. PTFs that are not marked as delayed are applied immediately or during the next IPL. PTFs marked as delayed are applied during the next IPL.
5. PTFs marked as delayed can only be applied permanently if they have been previously applied temporarily. Because they are delayed, these PTFs would be applied during the next scheduled IPL.
6. PTFs are always removed temporarily except Vertical Licensed Internal Code (VLIC) PTFs which are removed permanently.

Top

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACTIVITY	Activity	<i>Name, *GEN, *LAST</i>	Optional, Positional 2
ACTION	Action	<i>*SND, *RTV, *DLT, *APY, *SNDAPY, *RMV</i>	Optional, Positional 3
PTFID	PTF identifier	<i>Element list</i>	Optional
	Element 1: PTF	<i>Character value, *ALL</i>	
	Element 2: Product	<i>Character value, *ONLY</i>	
	Element 3: Release level	<i>Character value</i>	
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: <i>*NONE</i> Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CPNAME	Managed system node names	Single values: <i>*NONE</i> Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Communications name</i>	
PTFPART	PTF parts	<i>*PTF, *CVRLTR</i>	Optional
CVRLRLNG	Cover letter language	2900-2999, <i>*SRVRQS</i>	Optional
APY	Extent of change	<i>*TEMP, *PERM, *LODONLY</i>	Optional
DLYAPY	Delayed PTF apply	<i>*NO, *YES</i>	Optional
DLYRMV	Delayed PTF remove	<i>*NO, *YES</i>	Optional
RMTSTRTIME	Start time on managed system	<i>Element list</i>	Optional
	Element 1: Time zone	<i>*LCLSYS, *MGDSYS</i>	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time, *CURRENT</i>	
	Element 2: Date	<i>Date, *CURRENT, *NEXT</i>	
TEXT	Text 'description'	<i>Character value, *GEN, *BLANK</i>	Optional
COND	Activity conditions	Single values: <i>*NONE</i> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Activity	<i>Generic name, name, *PRV</i>	
	Element 2: Relational operator	<i>*EQ, *GT, *LT, *LE, *NE, *GE</i>	
	Element 3: End code	0-99, <i>*SUCCESS, *FAIL, *NOTRUN, *ANY</i>	
	Element 4: Condition mode	<i>*ALLNODES, *SAMENODE</i>	

Keyword	Description	Choices	Notes
STRTIME	Activity start time	<i>Element list</i>	Optional
	Element 1: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time, *CURRENT</i>	
	Element 2: Date	<i>Date, *CURRENT, *NEXT</i>	
	Element 2: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time, *ANY, *CURRENT</i>	
	Element 2: Date	<i>Date, *ANY, *CURRENT, *NEXT</i>	
HOLD	Hold activity	<i>*NO, *YES</i>	Optional

Top

---

## Change request description (CRQD)

Specifies the change request description object name.

The possible library values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched.

**\*CURLIB**

The current library for the job is used to locate the object.

*library-name*

Specify that only the library named in this parameter is searched.

The possible change request description value is:

*change-request-description*

Specify the name of the change request description object.

Top

---

## Activity (ACTIVITY)

Specifies the name of the activity to add to the change request description.

**\*GEN** An activity name is generated. The activity name is of the form QACTxxxxxx, where xxxxxx is the first multiple of ten not already being used.

**\*LAST**

The activity is the last to run in the change request. When \*LAST is specified for the activity (ACTIVITY) parameter, the condition (COND) parameter and the start time (STRTIME) parameter cannot be specified. Only one activity named \*LAST can exist in the change request description.

*activity-name*

Specify a 10-character activity identifier.

Top

---

## Action (ACTION)

Specifies the function to be performed on the PTF.

The possible values are:

- \***SND** Send the PTF to the specified managed system. The PTF is not loaded as part of this activity.
- \***RTV** Retrieve the specified PTF from the specified managed system.
- \***DLT** Deletes the PTF save file and cover letter on the specified managed system.
- \***APY** Apply the PTF on the specified system. All requisite PTFs must be applied. A PTF may set in loaded ("Not applied") status. If the PTF is not previously loaded, then the PTF must be loaded before it is applied. To apply all corequisite PTFs, set the status to "loaded". Request to apply one PTF and all corresponding corequisites will be applied.
- \***SNDAPY**  
Send, load, and apply the PTF on the specified system. All requisite PTFs must have already been applied. To apply the current PTF and corresponding corequisite PTFs, set the status to "loaded" ("Not applied"). When applying one PTF, that PTF and all corresponding corequisite PTFs are applied.
- \***RMV** Temporarily remove the PTF on the specified system.

Top

---

## PTF (PTFID)

Specifies the PTF that is to be distributed.

The possible values are:

### *PTF-identifier*

Specify a 7-character PTF identifier.

- \***ALL** All PTFs for the specified product. It can only be specified when applying or removing PTFs. A product must be specified.

The possible product identifier values are:

This is the product identifier of the product to which the PTF is associated.

### \*ONLY

The PTF identifier specified is associated with only one supported product.

### *product-ID*

Specify the product to which the PTF is associated. The product must be specified when PTF identifiers are not unique across products.

The possible release level of the product value is:

This element shows the release level of the product.

### *release(VxRxMx)*

Specify the release level of the product to which the PTF is associated. The format is VxRxMx. The release must be specified when PTF identifiers are not unique across product releases.

Top

---

## Managed systems node list (NODL)

Specifies that the node list parameter is the object name that contains a list of systems which are the destinations for the activity. This parameter cannot be specified if the control point name (CPNAME) parameter is also specified.

The possible value is:

**\*NONE**

The systems on which this activity is to be performed are not specified by a node list. Individual control point names must be specified.

The possible library values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

**\*CURLIB**

The current library for the job is used to locate the node list object.

*library-name*

Specify the name of the library to be searched.

The possible value is:

*node-list-name*

Specify the node list object name which contains the list of systems where the PTF function is to be performed.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if a node list (NODL) parameter is specified.

**\*NONE**

The systems on which this activity is performed are not identified individually. A node list must be specified.

The possible values are:

**\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

*network-identifier*

Specify the APPN network identifier of the managed system on which the activity is to be performed.

The possible control point name value is:

*control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed.

Top

---

## PTF parts (PTFPART)

Specifies whether the PTFs or cover letters should be sent, retrieved, or deleted.

**\*PTF** Only the PTF should be handled.

**\*CVRLTR**

Only the PTF cover letter is handled. \*CVRLTR can only be used with the \*RTV and \*SND actions.

If the delete action is specified, then both the PTF and the cover letter are always deleted.

Top

---

## Cover letter language (CVLTRLNG)

Specifies the language of the cover letter to be sent with the PTF.

**\*SRVRQS**

The cover letter language specified in the service requester entry for each node determines which cover letter that particular node is sent. If a service requester entry is not present or a language is not specified, the default language 2924 is used. If the language specified is not found, then 2924 is sent except when 2926 is requested in which case 2950 is sent.

*cover-letter-language-ID*

Specify the 4-character language identifier of the cover letter to be sent or retrieved.

Top

---

## Extent of change (APY)

Specifies the extent of change when the PTF is applied.

**\*TEMP**

The PTF is applied temporarily.

**\*PERM**

The PTF is applied permanently.

**\*LODONLY**

The PTF is only loaded. This is useful for PTFs which are part of a corequisite group where all PTFs are applied, removed, or permanently applied as a group and the "loaded" ("Not applied") status is the requisite to perform those functions.

Top

---

## Delayed PTF apply (DLYAPY)

Specifies how PTFs are applied. Apply immediate PTFs one at a time while the activity runs on the specified system or later during the next IPL.

**\*NO**

Applies an immediate PTF at the time the activity runs. If the PTF is marked delayed, it is not applied until the next unattended IPL.

**\*YES**

Applies both immediate or delayed PTFs during the next unattended IPL.

Top



---

## Delayed PTF remove (DLYRMV)

Specifies how a PTF is removed. Remove a PTF at the time the activity is specified on the system or later during the next IPL.

The possible single value is:

- \*NO Removes a PTF at the time an activity runs.
- \*Yes Removes during the next time the unattended IPL.

Top

---

## Start time on managed system (RMTSTRTIME)

Specifies the date and time when the activity can begin running on the managed system. The current date and time values, and the next date values are determined when the activity begins running at the central site system based on the central site date and time.

The possible time zone values are:

### \*LCLSYS

The remote start time is specified in the time zone of the central site system.

### \*MGDSYS

The remote start time is specified in the time zone of the managed system.

The possible start after time values are:

This is the definition of the time after which the activity is to start.

### \*CURRENT

This function can start on the managed system at any time on or after the time this activity is started on the central site system on the date specified in element 2.

### *start-after-time*

Specify the time when this function can start on the managed system. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where, hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

### \*CURRENT

This function starts on the managed system on any date on or after the activity starts on the central site system.

### \*NEXT

This function starts on the managed system on any date after the activity starts on the central site system.

### *start-after-date*

Specify the date after the functions start on the managed system. The date must be specified in the job date format.

## Notes

1. The special values \*CURRENT and \*NEXT are specified for the date and the time if the time zone value \*MGDSYS is specified.
2. This parameter can only be specified if \*APY, \*RMV, or \*SNDAPY actions are specified.

---

## Text 'description' (TEXT)

Specifies the activity description.

**\*GEN** A description is generated based on the activity selected.

*text-description*

Specify a 50 character description of the activity.

---

## Activity conditions (COND)

Specifies which conditions must be met before this activity can be performed. Each condition identifies an activity which must run before this activity and the value the end code from that activity which must have to allow this activity to run. The default condition is that the previous activity (in alphabetical order) must complete successfully before this activity can be run.

The possible single value is:

**\*NONE**

There are no conditions for this activity.

The possible conditioning activity values are:

The activity which must be run before this activity.

**\*PRV** This activity is conditioned on the previous activity. Activities are ordered alphabetically by activity name. If the activity being added is the first activity, a previous activity does not exist and any condition with \*PRV is marked as having been met.

*conditioning-activity-name*

Specify the name of the activity which must run before this activity. The activity name specified in the activity (ACTIVITY) parameter cannot be specified in the conditioning activity name. An activity cannot be conditioned on itself.

*generic\*-activity-name*

Specify the generic name of the activities which must run before this activity.

The possible relational operator values are:

This element is the relational operator to use when comparing the end code from the conditioning activity.

**\*EQ** Equal

**\*GT** Greater than

**\*LT** Less than

**\*NE** Not equal

**\*GE** Greater than or equal

**\*LE** Less than or equal

The possible condition code values are:

### \*SUCCESS

The activity ended successfully (0 <= end code <= 9). This end code can only be specified with relational operator \*EQ or \*NE.

**\*FAIL** The activity failed (10 <= end code <= 89). This end code can only be specified with relational operator \*EQ or \*NE.

### **\*NOTRUN**

The activity is never started (90 <= end code <= 99). This end code is only specified with relational operator \*EQ or \*NE.

**\*ANY** The activity ended with any end code. This end code is only specified with relational operator \*EQ.

### *end-code*

Specify an integer value (0-99) that indicates the result of an activity (success or failure). The end code ranges and descriptions are:

- 00** Activity completed successfully.
- 01-09** Activity completed with warning messages.
- 10-29** Activity did not complete successfully.
- 30-39** Activity was canceled by a user before it completed.
  - 30 = Activity ended with \*CNTRLD option
  - 35 = Activity ended with \*IMMED option
  - 39 = Activity ended with \*FRCFAIL option
- 40-49** Activity was not run due to errors detected by the application.
  - 40 = Activity not run for security reasons
- 90-99** Activity was not run because conditions or schedules were not met.
  - 95 = Scheduled start time expired
  - 99 = Conditions cannot be met

The possible condition mode values are:

This element indicated which systems the conditioning activity must have completed on before this activity can be performed.

### \*ALLNODES

The conditioning activity specified must complete on all nodes before this activity runs.

### **\*SAMENODE**

When the conditioning activity specified completes for a given node, the activity specified on the ACTIVITY parameter may run for that same node even though the conditioning activity specified may not have completed for all other nodes. In the case where this activity lists a node not in the conditioning activity, this activity can run for that node; the condition is ignored.

Top

---

## **Activity start time (STRTIME)**

Specifies the date and time when this activity can be started on the central site system. The current date and time values and next date values are determined when the change request is submitted.

The possible start after time values are:

### **\*CURRENT**

This activity can start any time on or after the time when the change request is submitted.

#### ***start-after-time***

Specify the time when this activity can start. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

### **\*CURRENT**

This activity can start on or after the date when the change request is submitted.

### **\*NEXT**

The activity can start on any date after the date when the change request is submitted.

#### ***start-after-date***

Specify the date after this activity can start. The date must be specified in the job date format.

The possible start before time values are:

This element is ignored if the start before date is \*ANY.

**\*ANY** The activity can start at any time on or before the start before date.

### **\*CURRENT**

The activity must start before the time when the change request was submitted on the date specified on the start before date element. This value cannot be specified if the start before date is \*CURRENT.

#### ***start-before-time***

Specify the time before which the activity must start. If the activity cannot be started before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start before date values are:

**ANY** The activity can start at any time after the start after time and the start after date.

### **\*CURRENT**

The activity must start on the date the change request is submitted.

### **\*NEXT**

The activity must start by the day after the date the change request is submitted.

#### ***start-before-date***

Specify the date before the activity must start. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

Top

---

## **Hold activity (HOLD)**

Specifies that the activity be held when the change request is submitted.

**\*NO** The activity is not held. It runs when all conditions and the start time are met.

**\*YES** The activity is held for all nodes when the change request is submitted. It must be released by you before it runs.

Top

---

## Examples

### Example 1: Adding an Activity to Permanently Apply a PTF

```
ADDPTFCRQA  CRQD(MYLIB/CR1)  ACTIVITY(ACT01)  ACTION(*APY)
             APY(*PERM)  PTFID(SF12345)
             CPNAME((*NETATR SYS1))
```

This command adds an activity to permanently apply PTF SF12345 on system SYS1.

### Example 2: Add an Activity to Send a PTF Cover Letter

```
ADDPTFCRQA  CRQD(MYLIB/CR2)  ACTIVITY(ACT03)  ACTION(*SND)
             PTFID(SF89345)  PTFPART(*CVLRLTR)  CVLRLRLNG(2950)
             NODL(NETLIB/EUROPE SYS)
```

This command adds an activity to send a PTF cover letter to all systems in Europe that are identified in the EUROPE SYS node list.

### Example 3: Adding an Activity to Temporarily Apply a PTF

```
ADDPTFCRQA  CRQD(MYLIB/CR3)  ACTIVITY(ACT01)  ACTION(*SND)
             PTFID(SF99911)  CPNAME((NET1 SYSX))
ADDPTFCRQA  CRQD(MYLIB/CR3)  ACTIVITY(ACT02)  ACTION(*APY)
             APY(*TEMP)  PTFID(SF99911)
             RMTSTRTIME(*MGDSYS ('02:00:00' '12/12/02'))
             CPNAME((NET1 SYSX))
```

This command adds activities to send a PTF and apply it temporarily at 2:00 a.m. on December 12th, 2002, on a system encountering a problem.

### Example 4: Adding an Activity to Load a PTF

```
ADDPTFCRQA  CRQD(MYLIB/CR4)  ACTIVITY(ACT01)  ACTION(*APY)
             PTFID(SF89555)  CPNAME(*NETATR SYS4)  APY(*LODONLY)
```

This command adds an activity to only load the PTF SF89555, which is part of a corequisite PTF group, on system SYS4.

### Example 5: Adding an Activity to Send and Permanently apply a PTF in Delayed Mode

```
ADDPTFCRQA  CRQD(MYLIB/CR5)  ACTIVITY(ACT01)  ACTION(*SNDAPY)
             PTFID(SF91388)  CPNAME(*NETATR SYS5)
             APY(*PERM)  DLYAPY(*YES)
```

This command adds an activity to send and permanently apply PTF SF91388 during the next unattended IPL.

### Example 6: Adding an Activity to Load and Apply an Individual PTF and the Corequisite Group

```
ADDPTFCRQA  CRQD(MYLIB/CR6)  ACTIVITY(ACT01)  ACTION(*SNDAPY)
             PTFID(SF97001)  CPNAME(*NETATR SYS6)
             APY(*LODONLY)
ADDPTFCRQA  CRQD(MYLIB/CR6)  ACTIVITY(ACT02)  ACTION(*SNDAPY)
             PTFID(SF97002)  CPNAME(*NETATR SYS6)
             APY(*LODONLY)
ADDPTFCRQA  CRQD(MYLIB/CR6)  ACTIVITY(ACT03)  ACTION(*SNDAPY)
             PTFID(SF97003)  CPNAME(*NETATR SYS6)
             APY(*TEMP)
```

In this example, there are three PTFs: SF97001, SF97002, and SF97003. SF97001 has corequisite SF97002, and SF97002 has corequisite SF97003. To apply one of them, you must apply all of them as a group to have the product function correctly. Therefore, you need to load the PTFs first, then when you request to apply only one of them, that PTF and its associated corequisites will all be applied.

[Top](#)

---

## **Error messages**

None

[Top](#)

## Add Resource CRQ Activity (ADDRSCCRQA)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Add Resource Change Request Activity (ADDRSCCRQA) command adds an activity to a change request description which performs an IPL of a managed iSeries system or restarts a non-iSeries resource such as a PS/2 with NetView DM/2 installed.

### Restrictions:

1. You must have \*CHANGE authority to the change request description and \*EXECUTE to the library.
2. A resource other than \*SYSx is not supported for iSeries managed systems.
3. If a value for NODL is specified, the node list can only contain entries that have a value of \*SNA for the address type.
4. Keylock must be in NORMAL position or the request is rejected.

### Notes:

The following notes provide information on how the command works.

1. Authorization to the product specified on the activity is not verified until the activity runs.
2. All conditions must be satisfied before the activity can run.
3. The start times indicate when the activity can be started. Actual start times can be later due to network and system delays.

Top

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACTIVITY	Activity	<i>Name, *GEN, *LAST</i>	Optional, Positional 2
ACTION	Action	*RESTART	Optional, Positional 3
RSC	Resource	<i>Name, *SYS, *SYSA, *SYSB</i>	Optional
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: *NONE Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CPNAME	Managed system node names	Single values: *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Character value</i>	

Keyword	Description	Choices	Notes
<b>OPTION</b>	How to end	<u>*CNTRLD</u> , *IMMED	Optional
<b>DELAY</b>	Delay time, if *CNTRLD	0-65535, <b>3600</b>	Optional
<b>TEXT</b>	Text 'description'	<i>Character value</i> , *GEN	Optional
<b>COND</b>	Activity conditions	Single values: *NONE Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Activity	<i>Generic name, name</i> , *PRV	
	Element 2: Relational operator	*EQ, *GT, *LT, *LE, *NE, *GE	
	Element 3: End code	0-99, *SUCCESS, *FAIL, *NOTRUN, *ANY	
	Element 4: Condition mode	*ALLNODES, *SAMENODE	
<b>STRTIME</b>	Activity start time	<i>Element list</i>	Optional
	Element 1: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *CURRENT	
	Element 2: Date	<i>Date</i> , *CURRENT, *NEXT	
	Element 2: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *ANY, *CURRENT	
	Element 2: Date	<i>Date</i> , *ANY, *CURRENT, *NEXT	
<b>RMTSTRTIME</b>	Start time on managed system	<i>Element list</i>	Optional
	Element 1: Time zone	*LCLSYS, *MGDSYS	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *CURRENT	
	Element 2: Date	<i>Date</i> , *CURRENT, *NEXT	
<b>HOLD</b>	Hold activity	*NO, *YES	Optional

Top

---

## Change request description (CRQD)

Specifies the change request description object name.

The possible library values are:

**\*LIBL** All of the libraries in the user and in the system portions of the job's library list are searched.

**\*CURLIB**

The current library for the job is used to locate the object.

*library-name*

Specify the name of the library to be searched.

The possible change request description value is:

*change-request-description*

Specify the name of the change request description object.

Top

---

## Activity (ACTIVITY)

Specifies the name of the activity to add to the change request description.



**\*GEN** An activity name is generated. The activity name is of the form QACTxxxxxx, where xxxxxx is the first multiple of ten not already being used.

**\*LAST**

The activity is the last to run in the change request. When \*LAST is specified for the activity (ACTIVITY) parameter, the condition (COND) parameter and the start time (STRTIME) parameter cannot be specified. Only one activity named \*LAST can exist in the change request description.

*activity-name*

Specify a 10-character activity name.

Top

---

## Action (ACTION)

Specifies the functions to be performed on the resource.

**\*RESTART**

Use to restart the specified resource. If the resource specified is \*SYS, \*SYSA, or \*SYSB, the managed system stops and restarts.

Top

---

## Resource (RSC)

Specifies the resource name.

**\*SYS** The action is performed against the entire system. For the restart action, this means that the managed system is powered down and restarted. The system panel determines the IPL source.

**\*SYSA**

This is the same as \*SYS except the IPL source is the A side.

**\*SYSB** This is the same as \*SYS except the IPL source is the B side.

*resource-name*

Specify a 16-character resource name on the managed system. This value is not supported by iSeries managed systems.

Top

---

## Managed systems node list (NODL)

Specifies that the node list parameter is the object name that contains a list of systems which are the destinations for the activity. This parameter cannot be specified if the control point name (CPNAME) parameter is also specified.

The possible value is:

**\*NONE**

The systems on which this activity is to be performed are not specified by a node list. Individual control point names must be specified.

The possible library values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

**\*CURLIB**

The current library for the job is used to locate the node list object.

***library-name***

Specify the name of the library to be searched.

The possible node list name value is:

***node-list-name***

Specify the node list object name containing the list of systems on which the activity is to be performed.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is performed. Control point names cannot be specified if a node list is specified.

**\*NONE**

The systems on which this activity is performed are not identified individually. A node list must be specified.

The possible values are:

**\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

***network-identifier***

Specify the APPN network identifier of the managed system on which the activity is performed.

The possible control point name value is:

***control-point-name***

Specify the APPN control point name of the managed system on which the activity is to be performed. For NetView Distribution Management Agents, the control point name is the change control client which supports numeric characters (0-9) in the first position of control point names that are valid in other platforms.

Top

---

## How to end (OPTION)

Specifies how to end. It specifies whether the system allows the active subsystems to end processing of active jobs in a controlled manner, or whether the system ends the job immediately.

**\*CNTRLD**

The jobs are ended in a controlled manner. If the jobs cannot be ended in a controlled manner during the delay period, they are ended immediately.

**\*IMMED**

Jobs are ended immediately.

Top

---

## Delay time, if \*CNTRLD (DELAY)

Specifies the number of seconds the system waits for the jobs to be ended in a controlled manner.

**3600** The system waits one hour before ending the jobs immediately.

### *delay-time*

Specify the delay time in seconds.

Top

---

## Text 'description' (TEXT)

Specifies the activity description.

**\*GEN** A description is generated.

### *text-description*

Specify a 50 character description of the activity.

Top

---

## Activity conditions (COND)

Specifies which conditions must be met before this activity can be performed. Each condition identifies an activity which must be run before this activity and the value the end code from that activity must have to allow this activity to run. The default condition is that the previous activity (in alphabetical order) must complete successfully before this activity can be run.

The possible single value is:

### **\*NONE**

There are no conditions for this activity.

The possible conditioning activity values are:

The activity which must be run before this activity.

**\*PRV** This activity is conditioned on the previous activity. Activities are ordered alphabetically by activity name. If the activity being added is the first activity, a previous activity does not exist and any condition with \*PRV is marked as having been met.

### *conditioning-activity-name*

Specify the name of the activity which must be run before this activity. The activity name specified in the activity (ACTIVITY) parameter cannot be specified in the conditioning activity name. An activity cannot be conditioned on itself.

### *generic\*-activity-name*

Specify the generic name of the activities which must run before this activity.

The possible relational operator values are:

This element is the relational operator to use when comparing the end code from the conditioning activity.

**\*EQ** Equal

**\*GT** Greater than

**\*LT** Less than

- \*NE Not equal
- \*GE Greater than or equal
- \*LE Less than or equal

The possible condition code values are:

**\*SUCCESS**

The activity ended successfully (0 <= end code <= 9). This end code can only be specified with relational operator \*EQ or \*NE.

\*FAIL The activity failed (10 <= end code <= 89). This end code can only be specified with relational operator \*EQ or \*NE.

**\*NOTRUN**

The activity is never started (90 <= end code <= 99). This end code is only specified with relational operator \*EQ or \*NE.

\*ANY The activity ended with any end code. This end code is only specified with the relational operator \*EQ.

***end-code***

Specify an integer value (0-99) that indicates the result of an activity (success or failure). The end code ranges and descriptions are:

- 00** Activity completed successfully.
- 01-09** Activity completed with warning messages.
- 10-29** Activity did not complete successfully.
- 30-39** Activity was canceled by a user before it completed.
  - 30 = Activity ended with \*CNTRLD option
  - 35 = Activity ended with \*IMMED option
  - 39 = Activity ended with \*FRCFAIL option
- 40-49** Activity was not run due to errors detected by the application.
  - 40 = Activity not run for security reasons
- 90-99** Activity was not run because conditions or schedules were not met.
  - 95 = Scheduled start time expired
  - 99 = Conditions cannot be met

The possible condition mode values are:

This element indicates which systems the conditioning activity must have completed on before this activity can be performed.

**\*ALLNODES**

The conditioning activity specified must complete on all nodes before this activity runs.

**\*SAMENODE**

When the conditioning activity specified completes for a given node, the activity specified on the ACTIVITY parameter may run for that same node even though the conditioning activity specified may not have completed for all other nodes. In the case where this activity lists a node not in the conditioning activity, this activity may run for that node; the condition is ignored.

Top

---

## Activity start time (STRTIME)

Specifies the date and time when this activity can be started on the central site system. The current date and time values and next date values are determined when the change request is submitted.

The possible start after time values are:

### \*CURRENT

This activity can start any time on or after the time when the change request is submitted.

### *start-after-time*

Specify the time when this activity can start. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

### \*CURRENT

This activity can start on or after the date on which the change request is submitted.

### \*NEXT

The activity can start on any date after the date when the change request is submitted.

### *start-after-date*

Specify the date after this activity can start. The date must be specified in the job date format.

The possible start before time values are:

This element is ignored if the start before date is \*ANY.

\*ANY The activity can start at any time on or before the start before date.

### \*CURRENT

The activity must start before the time when the change request was submitted on the date specified on the start before data element. This value cannot be specified if the start before date is \*CURRENT.

### *start-before-time*

Specify the time before the activity must start. If the activity cannot be started before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss), where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start before date values are:

\*ANY The activity can start at any time after the start after time and the start after date.

### \*CURRENT

The activity must start on the date the change request is submitted.

### \*NEXT

The activity must start by the day after the date the change request is submitted.

### *start-before-date*

Specify the date before the activity must start. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

Top

---

## Start time on managed system (RMTSTARTIME)

Specifies the date and time when the activity can begin running on the managed system. The current date and time values and next date values are determined when the activity begins running on the central site system based on the central site date and time.

The possible time zone values are:

### \*LCLSYS

The remote start time is specified in the time zone of the central site system.

### \*MGDSYS

The remote start time is specified in the time zone of the managed system.

The possible start after time values are:

This is the definition of the time when the activity is to start.

### \*CURRENT

This function can start on the managed system at any time on or after the time this activity is started on the central site system on the date specified in element 3.

#### *start-after-time*

Specify the time when this function can start on the managed system. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

### \*CURRENT

This function starts on the managed system on any date on or after the activity starts on the central site system.

### \*NEXT

This function starts on the managed system on any date after the activity starts on the central site system.

#### *start-after-date*

Specify the date after the functions start on the managed system. The date must be specified in the job date format.

#### **Notes**

1. The special values \*CURRENT and \*NEXT are specified for the date and the time if the time zone value \*MGDSYS is specified.
2. This parameter can only be specified if \*RUN or \*SNDRUN actions are specified.

Top

---

## Hold activity (HOLD)

Specifies that the activity be held when the change request is submitted.

\*NO The activity is not held and it runs when all conditions and the start time are met.

\*YES The activity is held for all nodes when the change request is submitted. It must be released by a user before it will run.

Top

---

## Examples

### Example 1: Adding an Activity

```
ADDRSCCRQA  CRQD(MYLIB/CR1)  ACTIVITY(ACT01)
             ACTION(*RESTART) RSC(*SYS)
             STRTIME((04:00:00 *NEXT))
             NODL(NETLIB/STORES)
```

This command adds an activity to re-IPL the store systems at 4:00 a.m. tomorrow. These store systems are identified in the STORES node list.

### Example 2: Performing an IPL

```
ADDRSCCRQA  CRQD(MYLIB/CR1)  ACTIVITY(ACT01)
             ACTION(*RESTART) RSC(*SYSA)
             CPNAME((*NETATR SYS1)) OPTION(*IMMED)
```

This command adds an activity to perform an IPL of the system SYS1 immediately to the A side.

[Top](#)

---

## Error messages

None

[Top](#)





# Apply Remote PTF (APYRMTPTF)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Apply Remote Program Temporary Fix (APYRMTPTF) command allows a service provider to remotely apply PTFs on the service requester's system. When using the APYRMTPTF command, you can request an IPL of the service requester's system. The apply PTF and IPL can be scheduled to occur at a later date and time.

## Note:

A change request is automatically submitted that can be viewed to determine the status of this command. You can use the Work with Submitted Change Request (WRKSBMCRQ) command to monitor the status.

## Restrictions:

1. The apply and IPL functions of this command are only supported when the service requester has Managed System Services installed.
2. PTFs that are not marked as delayed are applied immediately. PTFs marked as delayed are scheduled to be applied at the next IPL.
3. PTFs marked as delayed can only be applied permanently if they have been previously applied temporarily. Notice that an IPL would have been required to apply them since they are delayed.
4. Prerequisite PTFs of the same product must already be loaded on the service requester. However, prerequisite PTFs of another product must already be applied on the service requester. Corequisite PTFs of the same product and same option must already be loaded on the service requester. However, corequisite PTFs of another product or another option must already be applied on the service requester.
5. If a NODL value is specified, the node list can only contain entries that have a value of \*SNA for the address type.

Top

## Parameters

Keyword	Description	Choices	Notes
PTFID	PTF description	Values (up to 300 repetitions): <i>Element list</i>	Required, Positional 1
	Element 1: PTF identifier	<i>Character value</i> , *ALL	
	Element 2: Product	<i>Character value</i> , * <b>ONLY</b>	
	Element 3: Release level	<i>Character value</i>	
DESTSRVRQS	Destination service requester	Single values: *ALL, * <b>SELECT</b> , *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name</i> , *NETATR	
	Element 2: Control point	<i>Communications name</i>	
APY	Extent of change	* <b>TEMP</b> , *PERM	Optional
DLYAPY	Delayed PTF apply	* <b>NO</b> , *YES	Optional

Keyword	Description	Choices	Notes
NODL	Node list name	<i>Qualified object name</i>	Optional
	Qualifier 1: Node list name	<i>Name, *NONE</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
IPL	IPL remote system	<i>*NO, *YES</i>	Optional
PWRDWNOPT	Power down option	<i>*CNTRLD, *IMMED</i>	Optional
PWRDWNDLY	Power down delay	<i>1-65535, 3600</i>	Optional
RMTAPYTIME	Remote apply time	<i>Element list</i>	Optional
	Element 1: Time zone	<i>*LCLSYS, *MGDSYS</i>	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time, *CURRENT</i>	
	Element 2: Date	<i>Date, *CURRENT, *NEXT</i>	
RMTIPLTIME	Remote IPL time	<i>Element list</i>	Optional
	Element 1: Time zone	<i>*LCLSYS, *MGDSYS</i>	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time, *CURRENT</i>	
	Element 2: Date	<i>Date, *CURRENT, *NEXT</i>	

Top

---

## PTF description (PTFID)

Specifies the PTF that is to be applied. A maximum of 300 PTF identifiers can be specified.

The possible PTF identifier values are:

### *PTF-identifier*

Specify a 7-character PTF identifier.

**\*ALL** Applies all PTFs for the specified product.

The possible product identifier values are:

### \*ONLY

Specifies that the PTF identifier is associated with only one product.

### *product-id*

Specify the product of the PTF. The product must be specified when PTF identifiers are not unique across products or the PTF identifier is \*ALL.

The possible release level of product value is:

### *release(VxRxMx)*

Specify the release level of the product. The format is VxRxMx.

Top

---

## Destination service requester (DESTSRVRQS)

Specifies the service requester where this PTF function is to be performed. A single service requester or a list of service requesters can be specified.

The possible network identifier values are:

### **\*SELECT**

Shows a list of service requesters. From the list, you can select one or more service requesters.  
\*SELECT is not valid when the command is used in a batch environment.

**\*ALL** Specifies that this PTF function should occur on all of the service requesters that are defined for this service provider.

### **\*NONE**

The service requesters are identified in a node list object.

### **\*NETATR**

The remote network identifier of the service requester is the same as the remote network identifier of this system.

### ***remote-network-identifier***

Specify the remote network identifier of the service requester.

The possible control point value is:

This element is not valid when the network identifier is \*ALL, \*SELECT, or \*NONE.

### ***remote-control-point***

Specify the remote control point name of the service requester.

Top

---

## **Extent of change (APY)**

Specifies the extent of the change when the PTFs are applied.

The possible values are:

### **\*TEMP**

Apply the PTFs temporarily on the service requester.

### **\*PERM**

Apply the PTFs permanently on the service requester.

Top

---

## **Delayed PTF apply (DLYAPY)**

Specifies how PTFs are applied. Apply immediate PTFs at the time the activity runs on the specified system or later during the next IPL.

**\*NO** Applies an immediate PTF at the time the activity runs. If the PTF is marked delayed, it is not applied until the next unattended IPL.

**\*YES** Applies both immediate or delayed PTFs during the next unattended IPL.

Top

---

## **Node list name (NODL)**

Specifies the node list object name that contains a list of service requesters where this PTF function is to be performed.

The possible value is:

**\*NONE**

The service requesters are identified in the DESTSRVRQS parameter.

The possible library values are:

**\*LIBL** Searches all of the libraries in the user and system portions of the job's library list for the node list object.

**\*CURLIB**

Searches for the node list object in the library for the current job.

*library-name*

Specify the name of the library to be searched.

The possible node list name value is:

*node-list-name*

Specify the node list name which contains the list of service requesters where this PTF function is to be performed.

Top

---

## **IPL remote system (IPL)**

Specifies if an IPL of the service requester's system should occur.

The possible values are:

**\*NO** An IPL of the service requester's system should not occur.

**\*YES** An IPL of the service requester's system should occur after all of the PTFs listed on the command are successfully applied on the service requester.

Top

---

## **Power down option (PWRDWOPT)**

Specifies how to manage the end of active jobs during the IPL processing at service requester system. This parameter is valid only if IPL(\*YES) is specified.

**\*CNTRLD**

All jobs end when the time specified in Power Down Delay (PWRDWDLY) parameter expires. Any current jobs that are running perform an end of batch processing.

**\*IMMED**

Jobs end immediately.

Top

---

## **Power down delay (PWRDWDLY)**

This parameter is valid only if IPL(\*YES) and if PWRDWOPT(\*CNTRLD) are specified. Specifies the amount of time in seconds that the system allows a controlled end to be performed. If the end of job routines are not finished in the specified power down delay, any remaining jobs are ended immediately.

**3600** The amount of time in which to complete a controlled end of processing is limited to 3600 seconds.

### *delay-time*

Specify the maximum amount of delay time, in seconds, in which a controlled end can be performed. Valid values range from 1 through 65535 (65,535 seconds).

Top

---

## Remote apply time (RMTAPYTIME)

Specifies the date and the time when this PTF function can occur on the service requester. The current date and time values and the next date are determined, when this PTF function is processed on this system, based on this system's date and time.

The possible time zone values are:

### \*LCLSYS

Specifies the remote start time in the time zone of this system.

### \*MGDSYS

Specifies the remote start time in the time zone of the service requester.

The possible start after time values are:

### \*CURRENT

Starts this PTF function on the service requester system at any time on or after this PTF function request is processed on this system.

### *start-after-time*

Specify the time when this PTF function can be started on the service requester. The time can be entered as 4 or 6 digits, such as hhmm or hhmmss where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a separator. With a time separator, specify a string of 5 or 8 digits such as hh:mm:ss.

The possible start after date values are:

### \*CURRENT

Starts this PTF function on the service requester system on any date on or after the date this PTF function request is processed on this system.

### \*NEXT

Starts this PTF function on the service requester system on any date after the date this PTF function is processed on this system.

### *start-after-date*

Specify the date when this PTF function can start on the service requester system. The date must be specified in the job date format.

**Note:** The special values \*CURRENT and \*NEXT cannot be specified for the date and time if the time zone value \*MGDSYS is specified.

Top

---

## Remote IPL time (RMTIPLTIME)

Specifies the date and time when the IPL can occur on the service requester. The current date and time values and next date are determined, when this IPL request is processed on this system, based on this system's date and time.

The possible time zone values are:

### **\*LCLSYS**

Specifies the remote start time in the time zone of this system.

### **\*MGDSYS**

Specifies the remote start time in the time zone of the service requester's system.

The possible start after time values are:

### **\*CURRENT**

Starts the IPL on the service requester system at any time on or after this IPL request is processed on this system.

### ***start-after-time***

Specify the time when the IPL can be started on the service requester. The time can be entered as 4 or 6 digits, such as hhmm or hhmmss where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a separator. With a time separator, specify a string of 5 or 8 digits such as hh:mm:ss.

The possible start after date values are:

### **\*CURRENT**

Starts the IPL on the service requester system on any date on or after the date the IPL request is processed on this system.

### **\*NEXT**

Starts the IPL on the service requester system on any date after the date this IPL request is processed on this system.

### ***start-after-date***

Specify the date when the IPL can start on the service requester system. The date must be specified in the job date format.

**Note:** The special values \*CURRENT and \*NEXT cannot be specified for the date and time if the time zone value \*MGDSYS is specified.

Top

---

## **Examples**

### **Example 1: Applying a Temporary PTF**

```
APYRMTPTF  PTFID((123456 *ONLY V5R2M0))  DESTSRVRQS(*SELECT)
           NODL(SRVPVDLIB/SRVRQSLIST)
```

Apply a PTF temporarily on the selected service requesters system.

### **Example 2: Applying Permanent PTFs**

```
APYRMTPTF  PTFID((*ALL PRODUCT1 V5R2M0))  DESTSRVRQS(*NONE)
           NODL(SRVPVDLIB/SRVRQSLIST)  APY(*PERM)
           RMTAPYTIME((*MGDSYS (10:00:00 10/31/02))
           IPL(*YES)  RMTIPLTIME((*MGDSYS (01:00:00 11/01/02))
```

Apply all PTFs permanently for a specified product, and schedule when the apply and IPL should occur for all of the service requesters in the node list object.

This example applies the PTFs on the service requesters starting at 10:00 a.m. on October 31, 2002. If the PTFs applied successfully, the IPL on the remote system starts at 1:00 a.m. on November 1, 2002. The times specified are in the time zone of the remote system.

### **Example 3: Applying a Delayed PTF**

```
APYRMTPTF  PTFID((987654 *ONLY V5R2M0))  DESTSRVRQS(*SELECT)
            NODL(SRVPVDLIB/SRVRQSLIST)  DLYAPY(*YES)
            RMTAPYTIME((*MGDSYS (8:00:00 *CURRENT))
```

Apply a delayed PTF temporarily on the selected service requesters system, and schedule when the apply should occur for all of the service requesters in the node list object.

This example applies the PTFs on the service requester systems starting at 8:00 am on the current date. The time specified is in the time zone of the remote system.

#### Example 4: Doing an IPL after Applying a PTF

```
APYRMTPTF  PTFID((*ALL *ONLY V5R2M0))  DESTSRVRQS(*SELECT)
            NODL(SRVPVDLIB/SRVRQSLIST)  IPL(*YES)
            RMTIPLTIME((*LCLSYS (*CURRENT 12/24/02))
            PWRDWNPT(*CNTRL)  PWRDWNPLY(1800)
```

Apply all PTFs temporarily for a specified product, and schedule when the IPL should occur for all of the service requesters in the node list object.

This example applies the PTFs on the service requester systems. If the PTFs are applied successfully, the IPL on the remote system starts at the current time on December 24, 2002. The times specified are in the time zone of the local system.

#### Example 5: Applying a PTF with two Corequisite PTFs and one Prerequisite PTF

```
APYRMTPTF  PTFID((SF00001 *ONLY V5R2M0))  DESTSRVRQS(*SELECT)
            RMTAPYTIME(*LCLSYS (*CURRENT *CURRENT))  IPL(*NO)
```

Apply PTF SF00001. This PTF has two corequisite PTFs and one prerequisite PTF. PTFs SF00002 and SF00003 are corequisite PTFs for PTF SF00001. PTF SF00004 is a prerequisite for PTF SF00001. The system will remove PTFs SF00001, SF00002, SF00003, and SF00004. The system will determine all corequisite PTFs if they were not specified. No IPL will be performed, and this action applies the PTFs on the service requesters. The times specified are in the time zone of the local system.

This example applies the PTFs on the service requesters system starting at 8:00 am on the current date. The time specified is in the time zone of the remote system.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF358A

PTF &1-&2 cannot be applied or removed.

Top





# Change Command CRQ Activity (CHGCMDCRQA)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Change Command Change Request Activity (CHGCMDCRQA) command changes command activity in a change request description to run a command on one or more managed systems.

The activity can be conditioned so that it only runs after one or more other activities have completed (successfully or unsuccessfully). The activity can also be scheduled to run at a date and at a time in the future.

## Restrictions:

1. You must have \*CHANGE authority to the change request description and \*EXECUTE authority to the library.
2. If a node list (NODL) value is specified, the node list can only contain entries that have a value of \*SNA for the address type.

## Notes:

The following notes provide information on how the command works.

1. Authorization to the product specified on the activity is not verified until the activity runs.
2. All conditions must be satisfied before the activity can run.
3. The start times indicate when the activity can be started. Actual start times can be later due to network and system delays.

Top

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Optional, Key, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACTIVITY	Activity	<i>Name, *LAST</i>	Optional, Key, Positional 2
CMD	Command to run	<i>Command string</i>	Optional, Positional 3
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: *NONE, *SAME Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Keyword	Description	Choices	Notes
CPNAME	Managed system node names	Single values: *SAME, *LOCAL, *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name</i> , *NETATR	
	Element 2: Control point	<i>Communications name</i>	
RMTSTRTIME	Start time on managed system	<i>Element list</i>	Optional
	Element 1: Time zone	*SAME, *LCLSYS, *MGDSYS	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *CURRENT, *SAME	
	Element 2: Date	<i>Date</i> , *CURRENT, *NEXT, *SAME	
	Element 3: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *ANY, *CURRENT, *SAME	
	Element 2: Date	<i>Date</i> , *ANY, *CURRENT, *NEXT, *SAME	
RTNSPLF	Return spooled file	*YES, *NO, *FAIL, *SAME	Optional
USRPRF	User profile	<i>Simple name</i> , *NONE, *SAME	Optional
PASSWORD	Password	<i>Character value</i> , *USRPRF, *NONE, *SAME	Optional
ENCODE	Encode command	*SAME, *YES, *NO	Optional
TEXT	Text 'description'	<i>Character value</i> , *SAME, *GEN	Optional
COND	Activity conditions	Single values: *SAME, *NONE Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Activity	<i>Generic name, name</i> , *PRV	
	Element 2: Relational operator	*EQ, *GT, *LT, *LE, *NE, *GE	
	Element 3: End code	0-99, *SUCCESS, *FAIL, *NOTRUN, *ANY	
	Element 4: Condition mode	*ALLNODES, *SAMENODE	
STRTIME	Activity start time	<i>Element list</i>	Optional
	Element 1: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *SAME, *CURRENT	
	Element 2: Date	<i>Date</i> , *SAME, *CURRENT, *NEXT	
	Element 2: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *SAME, *ANY, *CURRENT	
	Element 2: Date	<i>Date</i> , *SAME, *ANY, *CURRENT, *NEXT	
HOLD	Hold activity	*SAME, *NO, *YES	Optional

Top

## Change request description (CRQD)

Specifies the change request description object name.

The possible library values are the following:

\*LIBL All of the libraries in the user and in the system portions of the job's library list are searched.

\*CURLIB

The current library for the job is used to locate the object.

*library-name*

Specify that only the library named in this parameter is searched.

The possible values are:

*change-request-description*

Specify the name of the change request description object.

Top

---

## Activity (ACTIVITY)

Specifies the name of the activity to change in the change request description.

**\*LAST**

This activity is the last to run in the change request. When \*LAST is specified for the activity (ACTIVITY) parameter, the condition (COND) parameter and the start time (STRTIME) parameter cannot be specified. Only one activity named \*LAST can exist in the change request description.

*activity-name*

Specify a 10-character activity name

Top

---

## Command to run (CMD)

Specifies the command to run. It can be any command that is run in batch. The command must follow OS/400 CL command format of 1 to 10 characters with the first character being in alphabetical order and the other nine characters alphanumeric.

*command-string*

Specify the command to run on the managed system.

Top

---

## Managed systems node list (NODL)

Specifies that the node list parameter is the object name that contains a list of systems which are the destinations for the activity. This parameter cannot be specified if the control point name (CPNAME) parameter is also specified.

The possible values are:

**\*SAME**

The value does not change.

**\*NONE**

The systems on which this activity is to be performed are not specified by a node list. Individual control point names must be specified.

The possible values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

**\*CURLIB**

The current library for the job is used to locate the node list object.

*library-name.*

Specify the name of the library to be searched.

*node-list-name*

Specify the node list object name containing the list of systems on which the activity is to be performed.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if a node list is specified.

The possible values are:

**\*SAME**

The value does not change.

**\*NONE**

The systems on which this activity is performed are not identified individually. A node list must be specified.

**\*LOCAL**

The local system is identified as the target system. If \*LOCAL is specified, the command is run on the local system. The remote command service is not used. Any spooled files created remain on the system.

The possible values are:

**\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

*network-identifier*

Specify the APPN network identifier of the managed system on which the activity is to be performed.

The possible value is:

*control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed.

Top

---

## Start time on managed system (RMTSTRTIME)

Specifies the date and time when the activity can begin running on the managed system. The current date and time values, and the next date values are determined when the activity begins running at the central site system based on the central site date and time.

The possible time zone of the remote start time values are:

**\*SAME**

The value does not change.

**\*LCLSYS**

The remote start time is specified in the time zone of the central site system.

**\*MGDSYS**

The remote start time is specified in the time zone of the managed system.

The possible start after time values are:

This is the definition of the time after which the activity is to start.

**\*SAME**

The value does not change.

**\*CURRENT**

This function can start on the managed system at any time on or after the time this activity was started on the central site system.

***start-after-time***

Specify the time after which this function can be started on the managed system. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where, hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This function can start on the managed system on any date on or after the date this activity starts on the central site system.

**\*NEXT**

This function can start on the managed system on any date after the date this activity starts on the central site system.

***start-after-date***

Specify the date after the functions can start on the managed system. The date must be specified in the job date format.

The possible start before time values are:

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time on or after the start time.

**\*CURRENT**

The activity must start before the time the change request was submitted on the date specified on the start before date element.

***start-before-time***

Specify the time before which the activity must be started. If the activity cannot be started before this time, then it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator (:). With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds.

The possible start before date values are:

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any date on or after the start date.

**\*CURRENT**

The activity must start before the date the change request was submitted.

### **\*NEXT**

The activity must start by the day after the date the change request was submitted.

### ***start-before-date***

Specify the date before which the activity must be started. If the activity cannot be started by this date, then it never starts. The date must be specified in the job date format.

### **Note:**

The special values \*CURRENT and \*NEXT cannot be specified for the date and the time if the time zone value \*MGDSYS is specified.

Top

---

## **Return spooled file (RTNSPLF)**

Specifies whether the output spooled file from the remote command is returned.

The possible values are:

### **\*SAME**

The value does not change.

**\*YES** The spooled files created from the remote command are returned from the remote system. The spooled files from all of the nodes are combined into one spooled file that can be viewed by displaying the command activity details of the change request.

**\*NO** The output data is not returned from the remote system.

**\*FAIL** The spooled file job log is returned from the remote system if the command fails when running.

Top

---

## **User profile (USRPRF)**

Specifies the user profile under which the command runs at the managed systems. If the managed system is running the NetView Remote Operations Agent/400 product, this parameter is ignored at the managed systems.

The possible values are:

### **\*SAME**

The value does not change.

### **\*NONE**

No user profile is specified. The default user profile is used on each managed system.

### ***user-profile***

Specify the name of the user profile.

Top

---

## **Password (PASSWORD)**

Specifies the password for the remote user profile.

- Managed Systems at releases prior to V5R1M0 only accept uppercased passwords up to 10 characters long. If a longer password is entered, SMU18A2 message with 100B0007 SNA sense code is returned, indicating that the request was rejected.
- Managed Systems at release V5R1M0 and above, running with QPWDLVL system value:

- 0 or 1 — truncate the received passwords to 10 characters.
- 2 or 3 — accept passwords up to 128 characters.

The possible values are:

**\*SAME**

The value does not change.

**\*NONE**

No password is specified.

**\*USRPRF**

The password is the same as the user profile.

***password***

Specify the password for the user profile.

Top

---

## Encode command (ENCODE)

Specifies whether the command, user profile, and password are encoded when sent to the managed systems. If the managed system is running NetView Remote Operations Agent/400 product, this parameter must be set to \*NO.

**\*SAME**

The value does not change.

**\*YES**

The command, user profile, and password are encoded when the request is sent to the managed systems. The managed system attribute for the remote command key must be specified on both the central site system and the managed system. This attribute can be changed using the Change Managed System Attributes (CHGMGDSYSA) command. The remote command key must be the same on the central site system and the managed system.

**\*NO**

The command, user profile, and password are not encoded when the request is sent to the managed systems.

Top

---

## Text 'description' (TEXT)

Specifies the activity description.

**\*SAME**

The value does not change.

**\*GEN**

A description is generated based on the action specified.

***text-description***

Specify a 50-character description of the activity.

Top

---

## Activity conditions (COND)

Specifies which conditions must be met before this activity can be performed. Each condition identifies an activity which must be run before this activity and the value the end codes from that activity must have to allow this activity to run. The default condition is that the previous activity (in alphabetical order) must complete successfully before this activity can be run.

The possible single value is:

**\*NONE**

There are no conditions for this activity.

**\*SAME**

The value does not change.

The possible conditioning activity values are:

This is the activity which must be run before this activity.

**\*PRV** This activity is conditioned on the previous activity. Activities are ordered alphabetically by activity name. If the activity being added is the first activity, a previous activity does not exist and any condition with \*PRV is marked as having been met.

***conditioning-activity-name***

Specify the name of the activity which must be run before this activity. The activity name specified in the activity (ACTIVITY) parameter cannot be specified in the conditioning activity name. An activity cannot be conditioned on itself.

***generic\*-activity-name***

Specify the generic name of the activities which must be run before this activity.

The possible relational operator values are:

This element is the relational operator to use when comparing the end code from the conditioning activity.

**\*EQ** Equal

**\*GT** Greater than

**\*LT** Less than

**\*NE** Not equal

**\*GE** Greater than or equal

**\*LE** Less than or equal

The possible conditioning code results are:

**\*SUCCESS**

The activity ended successfully (0 <= end code <= 9). This end code can only be specified with relational operator \*EQ or \*NE.

**\*FAIL** The activity failed (10 <= end code <= 89). This end code can only be specified with relational operator \*EQ or \*NE.

**\*NOTRUN**

The activity was never started (90 <= end code <= 99). This end code is only specified with relational operator \*EQ or \*NE.

**\*ANY** The activity ended with any end code. This end code is only specified with relational operator \*EQ.

***end-code***

Specify an integer value (0-99) that indicates the result of an activity (success or failure). The end code ranges and descriptions are:

**00** Activity completed successfully.

**01-09** Activity completed with warning messages.



- 10-29 Activity did not complete successfully.
- 30-39 Activity was canceled by a user before it completed.
  - 30 = Activity ended with \*CNTRLD option
  - 35 = Activity ended with \*IMMED option
  - 39 = Activity ended with \*FRCFAIL option
- 40-49 Activity was not run due to errors detected by the application.
  - 40 = Activity not run for security reasons
- 90-99 Activity was not run because conditions or schedules were not met.
  - 95 = Scheduled start time expired
  - 99 = Conditions cannot be met

The possible conditioning mode values are:

The condition mode indicates which systems the activity must have completed on or before this activity can be performed.

**\*ALLNODES**

The activity specified in this condition element must complete on all nodes before this activity can run.

**\*SAMENODE**

When the conditioning activity specified completes for a given node, the activity specified on the ACTIVITY parameter may run for that same node even though the conditioning activity specified may not have completed for all other nodes. In the case where this activity lists a node not in the conditioning activity, this activity may run for that node; the condition is ignored.

Top

## Activity start time (STRTIME)

Specifies the date and time when this activity can start on the central site system. The current date and time values and next date values are determined when the change request is submitted.

The possible start after time values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This activity can start any time on or after the time when the change request was submitted.

***start-after-time***

Specify the time when this activity can start. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This activity can start on or after the date on which the change request is submitted.

**\*NEXT**

The activity can start on any date after the date the change request is submitted.

***start-after-date***

Specify the date after this activity can start. The date must be specified in the job date format.

The possible start before time values are:

This element is ignored if the start before date is \*ANY.

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time on or before the start before date.

**\*CURRENT**

The activity must start before the time when the change request was submitted on the date specified on the start before date element. This value cannot be specified if the start before date is \*CURRENT.

***start-before-time***

Specify the time before the activity must start. If the activity cannot be started before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where h = hours, m = minutes, and s = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start before date values are:

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time after the start after time and the start after date.

**\*CURRENT**

The activity must start on the date the change request is submitted.

**\*NEXT**

The activity must start by the day after the date the change request is submitted.

***start-before-date***

Specify the date before the activity must start. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

Top

---

## Hold activity (HOLD)

Specifies that the activity be held when the change request is submitted.

**\*SAME**

The value does not change.

**\*NO** The activity is not held. It runs when all conditions and the start time are met.

**\*YES** The activity is held for all nodes when the change request is submitted. It must be released by you before it runs.

Top

---

## Examples

### Example 1: Changing an Activity

```
CHGCMDCRQA  CRQD(MYLIB/CR1)  ACTIVITY(ACT01)  CMD(STRSBS QCMN)
```

This command changes an activity that starts the QCMN subsystem.

### Example 2: Changing an Activity for a Node List

```
CHGCMDCRQA  CRQD(MYLIB/CR2)  ACTIVITY(ACT01)
             CMD(STRSBS QSNADS)
             NODL(MYLIB/STORES)  RTNSPLF(*FAIL)
```

This example shows how activity ACT01 runs the command to start the subsystem QSNADS on the systems identified in the STORES node list. The example also asks for the spooled file to be returned to the central site system if the command fails.

### Example 3: Changing an Activity for Two Systems

```
CHGCMDCRQA  CRQD(MYLIB/CR3)  ACTIVITY(ACT02)
             CMD(PRODLIB/RUNREPORTS)
             CPNAME((STORENET STOREA) (STORENET STOREB))
             RMTSTRTIME(*MGDSYS (19:00 11/20/02)
                       (*ANY *CURRENT))  USRPRF(REPORTOPER)
             PASSWORD(OPERPASS)  ENCODE(*YES)
```

This example shows how activity ACT02 runs the detail reports for STOREA and STOREB after 7:00 p.m. on the managed system. The reports are run with the REPORTOPER user profile.

[Top](#)

---

## Error messages

None

[Top](#)



---

## Change CRQ Activity (CHGCRQA)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Change Change Request Activity (CHGCRQA) command modifies a change management activity to a change request description. The change management action that is to be performed by the added activity depends on the value specified on the action parameter.

The activity can be conditioned so that it will only run after one or more other activities have completed (successfully or unsuccessfully). The activity can also be scheduled to run at a date and time in the future.

### Restrictions

- The user must have \*CHANGE authority to the change request description object and \*EXECUTE authority to the library.
- If a NODL value is specified, the node list can only contain entries that have a value of \*SNA for the address type.
- Global names must be 64 characters or less including one separator character between tokens.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Required, Key, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACTIVITY	Activity	<i>Name, *LAST</i>	Optional, Key, Positional 2
ACTION	Action	*ACP, *ACT, *DLT, *INS, *RMV, *RTV, *RUN, *SND, *SNDINS, *SNDRUN, *UNINS, * <u>SAME</u>	Optional, Positional 3

Keyword	Description	Choices	Notes
GLBNAME	Global name	Single values: <b>*SAME</b> Other values: <i>Element list</i>	Optional
	Element 1: Global name token 1	<i>Character value</i> , *NETID, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 2: Global name token 2	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 3: Global name token 3	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 4: Global name token 4	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 5: Global name token 5	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 6: Global name token 6	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 7: Global name token 7	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 8: Global name token 8	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 9: Global name token 9	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 10: Global name token 10	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
COMPNAME	Component name	Single values: <b>*SAME</b> Other values: <i>Element list</i>	Optional
	Element 1: Component name token 1	<i>Character value</i> , *NETID	
	Element 2: Component name token 2	<i>Character value</i> , *NETID, *CPNAME	
	Element 3: Component name token 3	<i>Character value</i> , *NETID, *CPNAME	
	Element 4: Component name token 4	<i>Character value</i> , *NETID, *CPNAME	
	Element 5: Component name token 5	<i>Character value</i> , *NETID, *CPNAME	
	Element 6: Component name token 6	<i>Character value</i> , *NETID, *CPNAME	
	Element 7: Component name token 7	<i>Character value</i> , *NETID, *CPNAME	
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: *NONE, <b>*SAME</b> Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
CPNAME	Managed system node names	Single values: *NONE, <b>*SAME</b> Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name</i> , *NETATR	
	Element 2: Control point	<i>Character value</i>	
TEXT	Text 'description'	<i>Character value</i> , *GEN, *BLANK, <b>*SAME</b>	Optional

Keyword	Description	Choices	Notes
ACTFRC	Activation force	Single values: <b>*SAME</b> , *YES Other values: <i>Element list</i>	Optional
	Element 1: Activation force	*NO	
	Element 2: Delay units	*SECONDS, *MINUTES, *HOURS	
	Element 3: Delay period	1-65535	
ACTUSEACT	Activation use on activate	*BOTH, *PROD, *LAST, *NONE, <b>*SAME</b>	Optional
CPRTYPE	Compression type	*ADAPTIVE, *SNA, *USER, *NONE, <b>*SAME</b>	Optional
SNACPRCHR	SNA compression character	<i>Hexadecimal value</i> , *BLANK	Optional
USRCPRINF	User compression information	<i>Element list</i>	Optional
	Element 1: User compression name	<i>Name</i>	
	Element 2: User parameters	<i>Character value</i>	
CPRSTGSTT	Compression storage state	*DECOMPRESS, *COMPRESS, *NONE, <b>*SAME</b>	Optional
CPRTFRSTT	Compression transfer state	*DECOMPRESS, *COMPRESS, *NONE, <b>*SAME</b>	Optional
ACTUSEINS	Activation use on install	*PROD, *TRIAL, <b>*SAME</b>	Optional
ALTACTCOMP	Alter active components	*ALLOWED, *NOTALLOWED, *NONE, <b>*SAME</b>	Optional
AUTOACP	Auto accept	Single values: <b>*SAME</b> , *NONE, *NO Other values: <i>Element list</i>	Optional
	Element 1: Auto accept	*YES, *DESIRED	
	Element 2: Delay days	0-255	
AUTORMV	Auto remove	*YES, *NO, *DESIRED, *NONE, <b>*SAME</b>	Optional
PRETEST	Pre test	*YES, *NO, *DESIRED, <b>*SAME</b>	Optional
POSTTEST	Post test	*YES, *NO, *DESIRED, <b>*SAME</b>	Optional
ALWRMV	Allow remove	*NO, *DESIRED, *YES, <b>*SAME</b>	Optional
PARM	Initiate parameters	Values (up to 40 repetitions): <i>Not restricted</i>	Optional
KEEPOBJ	Keep object	*YES, *NO, *NONE, <b>*SAME</b>	Optional
COREQCHGNL	Corequisite change name list	Single values: *NONE, <b>*SAME</b> Other values: <i>Element list</i>	Optional
	Element 1: Global name	Values (up to 6 repetitions): <i>Element list</i>	
	Element 1: Global name token 1	<i>Character value</i>	
	Element 2: Global name token 2	<i>Character value</i>	
	Element 3: Global name token 3	<i>Character value</i>	
	Element 4: Global name token 4	<i>Character value</i>	
	Element 5: Global name token 5	<i>Character value</i>	
	Element 6: Global name token 6	<i>Character value</i>	
	Element 7: Global name token 7	<i>Character value</i>	
	Element 8: Global name token 8	<i>Character value</i>	
	Element 9: Global name token 9	<i>Character value</i>	
	Element 10: Global name token 10	<i>Character value</i>	

Keyword	Description	Choices	Notes
REPLACE	Replace object	*YES, *ALLOWED, *NO, <u>*SAME</u>	Optional
TODLTNAME	To be deleted name	Single values: *NONE, <u>*SAME</u> Other values: <i>Element list</i>	Optional
	Element 1: To be deleted name token 1	<i>Character value</i> , *NETID, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 2: To be deleted name token 2	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 3: To be deleted name token 3	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 4: To be deleted name token 4	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 5: To be deleted name token 5	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 6: To be deleted name token 6	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 7: To be deleted name token 7	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 8: To be deleted name token 8	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 9: To be deleted name token 9	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
	Element 10: To be deleted name token 10	<i>Character value</i> , *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME	
REFLVL	Refresh level	<i>Character value</i> , <u>*SAME</u>	Optional
FRCUNINS	Force uninstall	*YES, *NO, <u>*SAME</u>	Optional
RMTSTRTIME	Start time on managed system	Single values: *NONE Other values: <i>Element list</i>	Optional
	Element 1: Time zone	*LCLSYS, *MGDSYS, <u>*SAME</u>	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *CURRENT, <u>*SAME</u>	
	Element 2: Date	<i>Date</i> , *CURRENT, *NEXT, <u>*SAME</u>	
COND	Activity conditions	Single values: *NONE, <u>*SAME</u> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Activity	<i>Generic name, name</i> , <u>*PRV</u>	
	Element 2: Relational operator	<u>*EQ</u> , *GT, *LT, *LE, *NE, *GE	
	Element 3: End code	0-99, <u>*SUCCESS</u> , *FAIL, *NOTRUN, *ANY	
	Element 4: Condition mode	<u>*ALLNODES</u> , *SAMENODE	
STRTIME	Activity start time	<i>Element list</i>	Optional
	Element 1: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *CURRENT, <u>*SAME</u>	
	Element 2: Date	<i>Date</i> , *CURRENT, *NEXT, <u>*SAME</u>	
	Element 2: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *ANY, *CURRENT, <u>*SAME</u>	
	Element 2: Date	<i>Date</i> , *ANY, *CURRENT, *NEXT, <u>*SAME</u>	
HOLD	Hold activity	*NO, *YES, <u>*SAME</u>	Optional

Top



---

## Change request description (CRQD)

Change request description object name.

The possible values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched.

**\*CURLIB**

The current library for the job is used to locate the object.

*library-name*

Only the library named in this parameter is searched.

The possible change request description value is:

*change-request-description*

The name of the change request description object.

Top

---

## Activity (ACTIVITY)

The name of the activity to change in the change request description.

**\*LAST**

The activity will be the last to run in the change request.

*activity-ID*

A 10-character activity name.

Top

---

## Action (ACTION)

Specifies the change management action to be performed by the added activity.

**\*SAME**

The action value does not change.

**\*ACP** Relinquishes resources at a managed system required to maintain removability of a change. This cancels the removability of a change previously installed in a removable manner. The resources released are, typically, unaltered versions of components affected by the change.

**\*ACT** Causes the managed system to activate all previously installed changes. Each managed system implements the activation in its own way (for example, the activity will perform an initial program load (IPL) of a managed iSeries system or activate a configuration on a PS/2 and cause the PS/2 to restart).

**\*DLT** Requests a delete action at one or more managed systems.

**\*INS** Installs the objects previously packaged for installation on the specified managed system or systems. You can install up to seven objects. Only installable objects can be installed.

The objects are treated as corequisites, which means that either all installations succeed or all do not. The activity will use an object and its corequisites, if any, to alter all components necessary to effect the change. The managed system can perform such alteration in a removable manner if required, so that a subsequent request (Remove) can return all those components to their original condition prior to the alteration. Also automatic removal or automatic acceptance are possible.

**\*RMV** This activity, when run at the managed system, returns (that is, removes) all components

previously altered in connection with a change to their condition prior to the installation of the change. This is possible only for changes previously installed in a removable manner.

- \*RTV** Retrieves an object identified by its global name, from a managed system or from another central site system, for storage at the central site system. To retrieve an object from more than one system, a global name with an \*ANY token is required, so that each retrieved object has a unique global name. Global names with unspecified tokens (\*ANY, \*HIGHEST, or \*LOWEST) are stored in the distribution repository when they are retrieved.
- \*RUN** Causes a program or procedure to be executed at one or more managed systems.
- \*SND** Sends an object from the central site system to one or more managed systems or to another central site system.
- \*SNDINS**  
Sends the objects previously packaged for installation at the central site system to one or more managed systems and installs the object at the managed system. Only objects identified by global names can be installed.
- \*SNDRUN**  
Sends an object from the central site system to one or more managed systems and runs the object at the managed system.
- \*UNINS**  
Removes the objects installed on the specified managed system or systems. Only objects installed through an installable object can be uninstalled.

Top

---

## Global name (GLBNAME)

Specifies a global name, which is a series of tokens that uniquely identify an object in an SNA network. The global name represents the name that is used to locate the appropriate catalog entry on both the central site system and the managed systems. The catalog entry specifies the object that is used on that system.

Special values in a token position indicate how to search for the object. By specifying \*ANY in a token position, the token is ignored when searching for the correct object.

The possible values are:

### \*SAME

The value does not change.

#### **Element 1: Token 1**

### **\*NETID**

The first global name token value is a network ID generated by the command from the network attributes. The network ID is determined by the current value of the LCLNETID network attribute value.

### *global-name-token*

Specify the first token of the global name. The first token is recommended to be the registered enterprise ID or network ID.

#### **Element 2-10: Token 2-10**

- \*ANY** Any token value matches when searching for the object where the action is performed. This is useful when retrieving objects for which some of the tokens in the global name are not known or vary between systems.

**\*HIGHEST**

The object with the highest token value has the action performed on it. The token must be ordered. This is useful when a token in a global name is used to indicate a different version of the object and you need to manipulate the object with the highest version level.

**\*LOWEST**

The object with the lowest token value has the action performed on it. The token must be ordered. This is useful when a token in a global name is used to indicate a different version of the object and you need to manipulate the object with the lowest version level.

**\*NETID**

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

**\*CPNAME**

The control point name of this system is used. The control point is determined by the current value of the LCLCPNAME network attribute value.

**\*SERVER:**

This token is stored within the change request activity with the value &SERVER, and is replaced by the short name of the change control server when the object is distributed.

**\*TARGET:**

This token is stored within the change request activity with the value &TARGET, and is replaced by the short name of the target when the object is distributed.

**\*MDDATE:**

This token is stored within the change request activity with the value &DATE, and is replaced when distributed by the date the object was last modified.

**\*MDTIME:**

This token is stored within the change request activity with the value &TIME, and is replaced when distributed by the time the object was last modified.

***global-name-token***

Specify one of a series of 1 to 16 character tokens that uniquely identify the object on which the action is to be performed. Characters A through Z and 0 through 9 can be used. Other special values (@, #, and \$) can be used for tokens that represent network IDs and system names.

**Note:** GLBNAME is valid only when ACTION(\*ACT) or ACTION(\*UNINS) is not specified.

Top

---

## Component name (COMPNAME)

Component name, which is the set of global name tokens previous to the REF, FIX, or UPD subtree indicator (STI) tokens. The component name is used to identify the installable object that must be uninstalled from the managed system. The maximum number of tokens allowed is 7. COMPNAME is valid only when ACTION(\*UNINS) is specified.

The only special value allowed for the first token is \*NETID.

### Element 1: Token 1

The possible values are:

**\*SAME**

The value does not change.

**\*NETID**

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

***component-name-token***

One of a series of 1 to 16 character tokens that uniquely identifies the object on which the action is to be performed. Characters A through Z and 0 through 9 can be used. Other special values (@, #, and \$) can be used for tokens that represent network IDs and system names.

**Elements 2-10: Tokens 2-10****\*NETID**

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

**\*CPNAME**

The control point name of this system is used. The network ID is determined by the current value of the LCLCPNAME network attribute value.

***component-name-token***

One of a series of 1 to 16 character tokens that uniquely identifies the object on which the action is to be performed. Characters A through Z and 0 through 9 can be used. Other special values (@, #, and \$) can be used for tokens that represent network IDs and system names.

Top

---

## Managed systems node list (NODL)

Specifies that the node list parameter is the object name that contains a list of systems that are the destinations for the activity. This parameter cannot be specified if the control point name (CPNAME) parameter is also specified.

The possible values are:

**\*SAME**

The value does not change.

**\*NONE**

The systems on which this activity is to be performed are not specified by a node list. Individual control point names must be specified.

The possible library values are one of the following:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

**\*CURLIB**

The current library for the job is used to locate the node list object.

***library-name***

Specify the name of the library to be searched.

***node-list-name***

Specify the node list object name containing the list of systems on which the activity is to be performed.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if the node list (NODL) parameter is specified.

The possible values are:

**\*SAME**

The value does not change.

**\*NONE**

The systems on which this activity is performed are not identified individually. A node list must be specified.

**\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

*network-identifier*

Specify the APPN network identifier of the managed system on which the activity is to be performed. For NetView Distribution Management Agents, the network identifier is the change control server name.

*control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed. For NetView Distribution Management Agents, the control point name is the change control client which supports numeric characters (0-9) in the first position of control point names that are valid in other platforms.

Top

---

## Text 'description' (TEXT)

Specifies the activity description.

**\*SAME**

The value does not change.

**\*BLANK**

No text is specified.

**\*GEN** A description is generated based on the action selected. The descriptions generated are:

- Accept object
- Restart the system
- Delete object
- Run object
- Install object
- Remove object
- Retrieve object
- Send object
- Send and run object
- Send and install object
- Uninstall object

*text-description*

Specify a 50-character description of the activity.

---

## Activation force (ACTFRC)

Specifies whether or not the managed system should proceed with the activation based on its quiesced state.

### \*SAME

Activation force values do not change.

### *activation force*

Specify the activation force value.

**\*NO** The managed system will not proceed with the activation if the quiesce check shows that the managed system is still active.

### *delay units*

Specify the unit of time on which the delay period is specified.

### \*SECONDS

The delay period will be specified in seconds.

### \*MINUTES

The delay period will be specified in minutes.

### \*HOURS

The delay period will be specified in hours.

### *delay period*

Specifies the maximum amount of time that the managed system may wait to quiesce (if not already quiesced) before taking the action specified:

**3600** The delay period default is 3600 seconds.

### *delay-period*

Delay period range is 1-65535.

**\*YES** The managed system will proceed with the activation even if the quiesce check shows that the managed system is still active. Delay units and delay period are ignored when activation force is **\*YES**.

**Note:** ACTFRC is valid only when ACTION(\*ACT) is specified.

---

## Activation use on activate (ACTUSEACT)

Specifies which components altered by changes will be used during the activation.

### \*SAME

Activation use on activate value does not change.

### **\*BOTH**

Both, trial and production version.

### **\*PROD**

Production version only.

**\*LAST**

Last used; either both trial and production or production only.

**\*NONE**

No activation use on activate is specified.

**Note:** ACTUSEACT is valid only when ACTION(\*ACT) is specified.

Top

---

## Compression type (CPRTYPE)

Specifies the compression algorithm and related information associated with the compression of a particular change object. When either or both the transfer state or store state parameter specify compressed, this parameter must be present and one (and only one) of the adaptive compression, SNA compression, or user compression algorithms may be requested.

**\*SAME**

Compression algorithm does not change.

**\*ADAPTIVE**

Specify whether or not adaptive compression pertains to the requested object. iSeries does not support \*ADAPTIVE.

**\*SNA** Specify whether or not SNA compression pertains to the requested object.

**\*USER**

Specify whether or not a named user compression pertains to the requested object.

**\*NONE**

No compression type is specified.

**Notes:**

1. CPRTYPE is valid only when ACTION(\*RTV), ACTION(\*SND), ACTION(\*SNDRUN) or ACTION(\*SNDINS) is specified
2. CPRTYPE cannot be specified when CMPSTGSTT and CPRTFRSTT are not specified.
3. CPRTYPE cannot be specified when CMPSTGSTT is \*DECOMPRESS and CPRTFRSTT is not specified.
4. CPRTYPE cannot be specified when CPRSTGSTT is not specified and CPRTFRSTT is \*DECOMPRESS.
5. CPRTYPE cannot be specified when CPRSTGSTT and CPRTFRSTT are \*DECOMPRESS.

Top

---

## SNA compression character (SNACPRCHR)

Specifies information about the SNA compression algorithm as it pertains to the requested object. If omitted, the implication is that SNA compression does not pertain to the requested object.

**\*SAME**

SNA compression does not change.

**\*BLANK**

When SNA compression is requested, it is optionally specified with the implied default being the (X'40') character. Otherwise, it is not specified.

**SNA-prime-character**

The prime compression character to be associated with the single control byte (SCB) used by the SNA compression algorithm. Valid values are '00'X - 'FF'X

**Note:** SNACPRCHR is valid only when ACTION(\*RTV), ACTION(\*SND), ACTION(\*SNDRUN), or ACTION(\*SNDINS) is specified.

Top

---

## User compression information (USRCPRINF)

Specifies information about a named user compression algorithm as it pertains to the requested object. If omitted, the implication is that user compression does not pertain to the requested object.

### \*SAME

User compression does not change.

### *user-compression-name*

The name of the user compression algorithm that pertains to the requested object. It should be specified when user compression is requested. Otherwise, it is not specified.

### *user-parameters*

User parameters that apply to the user compression algorithm named in the user compression name. It is optionally specified when user compression is requested. Otherwise it is not specified.

**Note:** USRCPRINF is valid only when ACTION(\*RTV), ACTION(\*SND), ACTION(\*SNDRUN) or ACTION(\*SNDINS) is specified.

Top

---

## Compression storage state (CPRSTGSTT)

Specifies whether or not the object should be stored in compressed format at the managed system.

### \*SAME

Compression store state value does not change.

### **\*DECOMPRESS**

Store the object in decompressed format at the managed system. If already decompressed when it arrives at the managed system, store it as received. Otherwise, decompress the object using the compression method used to catalog the object in the managed system before storing it.

### **\*COMPRESS**

Store the object in compressed format at the managed system. When this value is specified, the compression algorithm must also be specified. If already compressed when it arrives at the managed system, store the object as received. Otherwise, compress the object using the compression algorithm before storing it.

### **\*NONE**

No compression storage state is specified.

**Note:** CPRSTGSTT is valid only when ACTION(\*RTV), ACTION(\*SND), ACTION(\*SNDRUN) or ACTION(\*SNDINS) is specified.

Top

---

## Compression transfer state (CPRTFRSTT)

Specifies whether or not the object should be transferred to the managed system in compressed format.

### \*SAME

Compression transfer state value does not change.



**\*DECOMPRESS**

Transfer the object in decompressed format at the managed system. If already decompressed at the source, transfer the object as stored. Otherwise, decompress the object using the compression method used to catalog the object in the managed system before transferring it.

**\*COMPRESS**

Compress the object using the Compression Algorithm and transfer the object in compressed format to the managed system. When this value is specified the compression algorithm must also be specified. If already compressed using a different algorithm at the source, decompress the object using the compression method used to catalog the object in the managed system before transferring it.

**\*NONE**

No compression transfer state is specified.

**Note:** CPRTFRSTT is valid only when ACTION(\*RTV), ACTION(\*SND), ACTION(\*SNDRUN) or ACTION(\*SNDINS) is specified.

Top

---

**Activation use on install (ACTUSEINS)**

Specifies whether the component to be altered by the installation process will be trial version or production version. If activation use on install is \*TRIAL, it means the object should be installed in the trial area to be tested, then removability must be \*YES.

**\*SAME**

Activation use on activate value does not change.

**\*TRIAL**

Trial version only.

**\*PROD**

Production version only.

**Note:** ACTUSEINS is valid only when ACTION(\*INS) or ACTION(\*SNDINS) is specified.

Top

---

**Alter active components (ALTACTIONCOMP)**

Specifies whether or not the managed system is allowed to apply the component alterations to the active system; if not, then such action is to be deferred until the next activation. This parameter is only allowed for \*SNDINS action.

**\*SAME**

Alter active components value does not change.

**\*ALLOWED**

Managed system is allowed to apply the component alterations to the active system.

**\*NOTALLOWED**

Managed system is not allowed to apply the component alterations to the active system.

**\*NONE**

No alter active component is specified.

**Note:** ALTACTIONCOMP is valid only when ACTION(\*INS), ACTION(\*SNDINS), or ACTION(\*RMV) is specified.

---

## Auto accept (AUTOACP)

Specifies whether the managed system accepts objects automatically if installation and any test performed are successful, in order to release resources required to maintain removability as soon as possible. Like a separate Accept request, the managed system deletes the objects after successful automatic acceptance.

### \*SAME

Auto accept value does not change.

**\*NO** Do not perform automatic acceptance.

### **\*NONE**

No auto accept is specified.

### *auto accept*

Auto acceptance possible values:

### **\*DESIRED**

Perform automatic acceptance if the specified managed system supports it.

**\*YES** Perform automatic acceptance.

### *delay-days*

Specifies the number of days the entry point is expected to wait before accepting the object automatically. Valid range of days is 0-255.

**Note:** AUTOACP is valid only when ACTION(\*INS) or ACTION(\*SNDINS) is specified.

---

## Auto remove (AUTORMV)

Specifies whether or not the managed system removes the object automatically if either installation or a test fails.

### \*SAME

Auto remove value does not change.

### **\*DESIRED**

Perform automatic removal if the specified managed system supports it.

**\*YES** Perform automatic removal.

**\*NO** Do not perform automatic removal.

### **\*NONE**

No auto remove is specified.

**Note:** AUTORMV is valid only when ACTION(\*INS) or ACTION(\*SNDINS) is specified.

---

## Pre test (PRETEST)

Specifies whether or not the entry point is to perform a test on the objects prior to installing them.

**\*SAME**

Pretest value does not change.

**\*DESIRED**

Perform a pretest, if the specified managed system supports it.

**\*YES** Perform a pretest.

**\*NO** Do not perform a pretest.

**Note:** PRETEST is valid only when ACTION(\*INS) or ACTION(\*SNDINS) is specified.

Top

---

## Post test (POSTTEST)

Specifies whether or not the entry point is to perform a test on the objects after installing or removing them.

**\*SAME**

Posttest value does not change.

**\*YES** Perform a posttest.

**\*DESIRED**

Perform a posttest if the specified managed system supports it.

**\*NO** Do not perform a posttest.

**Note:** POSTTEST is valid only when ACTION(\*INS), ACTION(\*SNDINS) or ACTION(\*RMV) is specified.

Top

---

## Allow remove (ALWRMV)

Specifies whether or not objects are to be installed in a removable manner (so that a subsequent Remove action can be issued against them).

**\*SAME**

Removability value does not change.

**\*YES** Install the object in a removable manner.

**\*DESIRED**

Install the object in a removable manner if the specified managed system supports it.

**\*NO** Do not install the object in a removable manner.

**Note:** ALWRMV is valid only when ACTION(\*INS) or ACTION(\*SNDINS) is specified.

Top

---

## Initiate parameters (PARM)

Specifies the parameters to be passed when initiating an object.

**\*SAME**

Parameter value does not change.

***parameter data***

Specifies the parameters to be passed when starting the program. Each parameter is a 1 to 253

character. The prompt panel will initially allow 25 characters to be entered. By entering an & in position 1, the field will be expanded for larger parameters. If the parameters include blanks or special characters enclose them in apostrophes.

**Note:** PARM is valid only when ACTION(\*RUN) or ACTION(\*SNDRUN) is specified.

Top

---

## Keep object (KEEPOBJ)

Specifies whether or not the object should be kept or deleted after the function has been successfully performed.

### \*SAME

Object disposition value does not change.

**\*YES** The object should be kept after performing the function.

**\*NO** The object should be deleted after performing the function.

### **\*NONE**

No keep object is specified.

**Note:** KEEPOBJ is valid only when ACTION(\*RUN), ACTION(\*INS), ACTION(\*SNDRUN) or ACTION(\*SNDINS) is specified.

Top

---

## Corequisite change name list (COREQCHGNL)

Specifies a list of SNA/file services (SNA/FS) global names identifying the names of objects that are to be installed by the entry point as a part of the installation of the object to be retrieved. A maximum of six corequisite change names are allowed.

The global name is a unique name you assign to the object so that it is not confused with any other object in a network. The global name represents the name that will be used to locate the appropriate catalog entry on both the central site system and managed systems.

### \*SAME

Corequisite change name list does not change.

### **\*NONE**

No corequisite change name list is specified.

### *global-name-token*

One of a series of 1-16 character tokens that uniquely identify the object on which the action is to be performed. Characters A through Z and 0 through 9 may be used. Other special values (@, #, and \$) may be used for tokens that represent network IDs and system names.

**Note:** COREQCHGNL is valid only when ACTION(\*INS) or ACTION(\*SNDINS) is specified.

Top

---

## Replace object (REPLACE)

Specifies that the object should be replaced if it already exists.

**\*SAME**

Replace value does not change.

**\*NO** The object must be added.

**\*YES** The object must be replaced.

**\*ALLOWED**

The object should be replaced or added.

**Note:** REPLACE is valid only when ACTION(\*RTV), ACTION(\*SND), ACTION(\*SNDRUN) or ACTION(\*SNDINS) is specified.

Top

---

## To be deleted name (TODLTNAME)

Specifies the name of the object, at the managed system location, that is to be deleted.

Special values in a token position indicate how to search for the object. By specifying \*ANY in a token position, it means that the token is ignored when searching for the correct object. If multiple objects are found matching the tokens specified an error is returned.

The single values allowed are:

**\*SAME**

The to be deleted name does not change.

**\*NONE**

No to be deleted name is specified.

The special values allowed for the first token are \*NETID, \*SERVER, \*TARGET, \*MDDATE, and \*MDTIME.

The special values allowed for the other tokens are:

**\*ANY** Any token value matches when searching for the object upon which the action is to be performed.

**\*HIGHEST**

The object with the highest token value will have the action performed upon it. The token must be ordered. Useful when a token in a global name is used to indicate a different version of the object and you wish to manipulate the object with the highest version level.

**\*LOWEST**

The object with the lowest token value will have the action performed upon it. The token must be ordered. Useful when a token in a global name is used to indicate a different version of the object and you wish to manipulate the object with the lowest version level.

**\*NETID**

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

**\*CPNAME**

The control point name of this system is used. The network ID is determined by the current value of the LCLCPNAME network attribute value.

**\*SERVER:**

This token is stored within the change request activity with the value &SERVER, and is replaced by the short name of the change control server when the object is distributed.

**\*TARGET:**

This token is stored within the change request activity with the value &TARGET, and is replaced by the short name of the target when the object is distributed.

**\*MDDATE:**

This token is stored within the change request activity with the value &DATE, and is replaced when distributed by the date the object was last modified.

**\*MDTIME:**

This token is stored within the change request activity with the value &TIME, and is replaced when distributed by the time the object was last modified.

***global-name-token***

One of a series of 1-16 character tokens that uniquely identify the object on which the action is to be performed. Characters A-Z and 0-9 may be used. Other special values (@, #, and \$) may be used for tokens that represent network IDs and system names.

**Notes:**

1. TODLTNAME is valid only when ACTION(\*SND), ACTION(\*SNDRUN) or ACTION(\*SNDINS) is specified.
2. The number of tokens specified in the TODLTNAME parameter must match the number of tokens specified in the GLBNAME parameter.

Top

---

## Refresh level (REFLVL)

Specifies the level of the software component to be uninstalled.

**\*SAME**

Refresh level value does not change.

**\*NONE**

No refresh level list is specified.

***refresh-level***

Specify a numeric character string of up to 16 digits.

**Note:** REFLVL is valid only when ACTION(\*UNINS) is specified.

Top

---

## Force uninstall (FRCUNINS)

Specifies whether to perform the uninstallation even if the software component includes an object that is waiting to be installed or an object that is currently being installed.

**\*SAME**

Force uninstallation value does not change.

**\*NO**

Do not allow uninstallation if there are objects, for the software component, waiting to be installed.

**\*YES**

Allow uninstallation even if there are objects, for the software component, waiting to be installed.

**Note:** FRCUNINS is valid only when ACTION(\*UNINS) is specified.

Top

---

## Start time on managed system (RMTSTRTIME)

Specifies the date and time when the activity can begin running on the managed system. The current date and time values and the next date values are determined when the activity begins running at the central site systems based on the central site date and time.

The possible single value is:

**\*NONE**

There is no remote start time specified for the managed system.

### Element 1: Time Zone

The time zone of the remote start time.

The possible values are:

**\*SAME**

The value does not change.

**\*LCLSYS**

The remote start time is specified in the time zone of the central site system.

**\*MGDSYS**

The remote start time is specified in the time zone of the managed system.

### Element 2: Start After Time

This is the definition of the time after which the activity is to start.

**\*SAME**

The value does not change.

**\*CURRENT**

This function can start on the managed system at any time on or after the time this activity is started on the central site system on the date specified in element 2.

*start-after-time*

Specify the time when this function can start on the managed system. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

### Element 3: Start After Date

This is the start after date.

**\*SAME**

The value does not change.

**\*CURRENT**

This function starts on the managed system on any date on or after the activity starts on the central site system.

**\*NEXT**

This function starts on the managed system on any date after the activity starts on the central site system.

*start-after-date*

Specify the date after the functions start on the managed system. The date must be specified in the job date format.

**Notes:**

1. The special values \*CURRENT and \*NEXT cannot be specified for the date and the time when the time zone value \*MGDSYS is specified.
2. This parameter is valid when these actions are specified: \*ACP, \*ACT, \*RUN, \*INS, \*RMV, \*SNDRUN, \*SNDINS, or \*UNINS.

Top

---

## Activity conditions (COND)

Specifies which conditions must be met before this activity can be performed. Each condition identifies an activity that must run before this activity and the value the end code from that activity must have to allow this activity to run. The default condition is that the previous activity (in alphabetical order) must complete successfully before this activity can be run.

The possible single values are:

### \*SAME

The value does not change.

### \*NONE

There are no conditions for this activity.

### Element 1: Conditioning Activity

The activity that must be run before this activity.

**\*PRV** This activity is conditioned on the previous activity. Activities are ordered alphabetically by activity name. If the activity being added is the first activity, a previous activity does not exist and any condition with \*PRV is marked as having been met.

#### *conditioning-activity-name*

Specify the name of the activity that must be run before this activity. The activity name specified in the activity (ACTIVITY) parameter cannot be specified in the conditioning activity name. An activity cannot be conditioned on itself.

#### *generic\*-conditioning-activity-name*

Specify the generic name of the activities that must run before this activity.

### Element 2: Relational Operator

This element is the relational operator to use when comparing the end code from an activity.

**\*EQ** Equal

**\*GT** Greater than

**\*LT** Less than

**\*NE** Not equal

**\*GE** Greater than or equal

**\*LE** Less than or equal

### Element 3: Condition Code

The element is the value compared to the actual end code of the conditioning activity.

### \*SUCCESS

The activity ended successfully (0 <= end code <= 9). This end code can only be specified with relational operator \*EQ or \*NE.



**\*FAIL** The activity failed (10 <= end code <= 89). This end code can only be specified with relational operator \*EQ or \*NE.

**\*NOTRUN**

The activity never started (90 <= end code <= 99). This end code is only specified with relational operator \*EQ or \*NE.

**\*ANY** The activity ended with any end code. This end code is only specified with relational operator \*EQ.

**end-code**

Specify an integer value (0-99) that indicates the result of an activity (success or failure). The end code ranges and descriptions are:

**00** Activity completed successfully.

**01-09** Activity completed with warning messages.

**10-29** Activity did not complete successfully.

**30-39** Activity was canceled by a user before it completed.

- 30 = Activity ended with \*CNTRLD option
- 35 = Activity ended with \*IMMED option
- 39 = Activity ended with \*FRCFAIL option

**40-49** Activity was not run due to errors detected by the application.

- 40 = Activity not run for security reasons

**90-99** Activity was not run because conditions or schedules were not met.

- 95 = Scheduled start time expired
- 99 = Conditions cannot be met

**Element 4: Condition Mode**

This element indicates which systems the conditioning activity must have completed on before this activity can be performed.

**\*ALLNODES**

The conditioning activity specified must complete on all nodes before this activity runs.

**\*SAMENODE**

When the conditioning activity specified completes for a given node, the activity specified on the ACTIVITY parameter can run for that same node even though the conditioning activity specified cannot have completed for all other nodes. In the case where this activity can run for that node, the condition is ignored.

Top

---

## Activity start time (STRTIME)

Specifies the date and the time when this activity can be started on the central site system. The current date and time values and the next date values are determined when the change request is submitted.

**Element 1: Start After Time**

The possible values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This activity can start on or after the time when the change request is submitted.

***start-after-time***

Specify the time when this activity can start. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

**Element 2: Start After Date**

**\*SAME**

The value does not change.

**\*CURRENT**

This activity can start on or after the date on which the change request is submitted.

**\*NEXT**

The activity can start on any date after the date the change request is submitted.

***start-after-date***

Specify the date after this activity can start. The date must be specified in the job date format.

**Element 3: Start Before Time**

This element is ignored if the start before date is \*ANY.

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time on or before the start before date.

**\*CURRENT**

The activity must start before the time at which the change request was submitted on the date specified on the start before data element. This value cannot be specified if the start before date is \*CURRENT.

***start-before-time***

Specify the time before which the activity must start. If the activity cannot be started before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

**Element 4: Start Before Date**

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time after the start after time and the start after date.

**\*CURRENT**

The activity must start on the date the change request is submitted.

**\*NEXT**

The activity must start by the day after the date the change request is submitted.

***start-before-date***

Specify the date before the activity must start. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

Top

---

## Hold activity (HOLD)

Hold the activity when the change request is submitted.

### \*SAME

The value does not change.

**\*NO** The activity is not held and will run when all conditions are met.

**\*YES** The activity is held for all nodes when the change request is submitted. It must be released by you before it runs.

Top

---

## Examples

The following examples illustrate how to use the CHGCRQA command to schedule activities to be performed on managed systems controlled by a NetView Distribution Manager/6000 change control server. The examples shown here are grouped by their activity actions:

- Accept
- Activate
- Delete
- Run
- Install
- Remove
- Retrieve
- Send, Send and Run, and Send and Install
- Uninstall

### Example 1: Accept Actions

The examples shown here match the first two examples shown for the install examples and demonstrate how some of the installed objects can be accepted.

- The install activity added to the change request description is:

```
ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT01)
          ACTION(*INS)  GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))  ACTUSEINS(*PROD)
          ALTACTCOMP(*NOTALLOWED)  AUTOACP((*YES))
          ALWRMV(*YES)
          COREQCHGNL((REXX PROC UPDATE CONFIG))
```

Instead of installing the objects on the single system MARYPWS1, the objects should be installed on all the managed systems controlled by the change control server ROMSERV1. (All the managed systems are listed in the node list ROMCLIENTS stored in library ROMLIB.)

The activity is changed in the following example so the objects can be installed in a removable manner, if removability is supported on the managed systems.

```
CHGCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT01)
          ACTION(*SAME)  GLBNAME(*SAME)
          NODL(ROMLIB/ROMCLIENTS)  CPNAME(*NONE)
          ACTUSEINS(*SAME)  ALTACTCOMP(*SAME)
          AUTOACP((*NO))  ALWRMV(*DESIRED)
          COREQCHGNL(*SAME)
```

The objects IBM 1234567 PMGRAB UPD 1 2 and REXX PROC UPDATE CONFIG are installed in a removable manner. The objects can be accepted by changing the accept activities. The activities were added to the change request as follows:

```

ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(ACCACT01)
          ACTION(*ACP)  GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          COND((INSACT03 *EQ 20 *SAMENODE))
ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(ACCACT02)
          ACTION(*ACP)  GLBNAME(REXX PROC UPDATE CONFIG)
          CPNAME((ROMSERV1 MARYPWS1))
          COND((INSACT03 *EQ 20 *SAMENODE))

```

But the activities need to be changed to:

```

CHGCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(ACCACT01)
          ACTION(*SAME)  GLBNAME(*SAME)
          NODL(ROMLIB/ROMCLIENTS)  CPNAME(*NONE)
          COND((INSACT03 *EQ *SUCCESS *ALLNODES))
CHGCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(ACCACT02)
          ACTION(*SAME)  GLBNAME(*SAME)
          NODL(ROMLIB/ROMCLIENTS)  CPNAME(*NONE)
          COND((INSACT03 *EQ *SUCCESS *ALLNODES))

```

- The install activity added to the change request description is:

```

ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT02)
          ACTION(*INS)  GLBNAME(EURO WORDPROD UPD 2 3 US)
          CPNAME((EUROITAL FREDSSWS))  ALWRMV(*YES)

```

Instead of installing the objects on the single system MARYPWS1, the objects should be installed on all the managed systems controlled by the change control server EUROITAL. (All the managed systems are listed in the node list EUROCLIENT stored in library EUROLIB.) To install the objects on all managed systems, the activity is changed to:

```

CHGCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT02)
          ACTION(*SAME)  GLBNAME(*SAME)
          NODL(EUROLIB/EUROCLIENT)  CPNAME(*NONE)
          ALWRMV(*YES)

```

To accept the object EURO WORDPROD UPD 2 3 US on all the managed systems, the activity ACCACT01 is added to the change request CRQACPRMV:

```

ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(ACCACT01)
          ACTION(*ACP)  GLBNAME(EURO WORDPROD UPD 2 3 US)
          CPNAME((EUROITAL FREDSSWS))
          COND((INSACT05 *EQ *SUCCESS *SAMENODE))

```

Should be changed as follows:

```

CHGCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(ACCACT01)
          ACTION(*SAME)  GLBNAME(*SAME)
          NODL(EUROLIB/EUROCLIENT)  CPNAME(*NONE)
          COND((INSACT05 *EQ *SUCCESS *ALLNODES))

```

## Example 2: Activate Actions

- Change an activate activity to use trial a production version during the activation.

```

CHGCRQA  CRQD(CCLIB/CRQINSACT)  ACTIVITY(ACTACT01)
          ACTION(*ACT)  CPNAME((ROMSERV1 MARYPWS1))
          ACTFRC(*NO *HOURS 6)  ACTUSEACT(*BOTH)
          STRTIME((2:00:00 *NEXT)(8:00:00 *NEXT))

```

The activation will occur after 2 a.m. but before 8 a.m. the next morning the request is submitted. The MARYPWS1 system will wait up to six hours before proceeding with the activation if the system is still active.

- Change an activity to schedule the activation of all previously installed objects on managed system FREDSSWS. FREDSSWS is controlled by the change control server EUROITAL. The activation will occur as soon as possible after 11 p.m. on 15 April 2002 in the time zone where FREDSSWS is located. The activation will occur even if the managed system FREDSSWS is still active.

The delay units and delay period are ignored because activation force is \*YES.

```

CHGCRQA  CRQD(CCLIB/CRQINSACT)  ACTIVITY(ACTACT02)
          ACTION(*ACT)  CPNAME((EUROITAL FREDSSWS))
          ACTFRC((*YES *MINUTES 30))
          RMTSTRIME(*MGDSYS (23:00:00 04/15/02))

```

### Example 3: Delete Actions

- Change an activity to delete the object EURO WORDPROD UPD 2 3 US on managed system FREDSW. FREDSW is controlled by the change control server EUROITAL.  
CHGCRQA CRQD(CCLIB/CRQDLT) ACTIVITY(DLTACT01)  
ACTION(\*DLT) GLBNAME(EURO WORDPROD UPD 2 3 US)  
CPNAME((EUROITAL FREDSW))
- Change an activity to delete all test files from systems MARYPWS1, MARYPWS2, MARYPWS3, and MARYPWS4. All these systems are controlled by the change control server ROMSERV1. The test files on those systems are cataloged as EURO SPELLCHECK TEST file-name.

```
CHGCRQA CRQD(CCLIB/CRQSNDLT) ACTIVITY(DLTACT02)
ACTION(*ACT)
GLBNAME(EURO SPELLCHECK TEST *ANY)
CPNAME((ROMSERV1 MARYPWS1) (ROMSERV1 MARYPWS2)
        (ROMSERV1 MARYPWS3) (ROMSERV1 MARYPWS4))
COND((SNDACT01 *EQ *SUCCESS *ALLNODES))
```

This activity occurs only if the object EURO SPELLCHECK EXE 1 US is successfully sent to those systems:

```
ADDCRQA CRQD(CCLIB/CRQSNDLT) ACTIVITY(SNDACT01)
ACTION(*SND) GLBNAME(EURO SPELLCHECK EXE 1 US)
CPNAME((ROMSERV1 MARYPWS1) (ROMSERV1 MARYPWS2)
        (ROMSERV1 MARYPWS3) (ROMSERV1 MARYPWS4))
REPLACE(*ALLOWED)
```

### Example 5: Run Actions

- Change an activity to run the program or script known by the global name EURO.VIRUSCHK.EXE.1.US on managed system FREDSW. Pass the parameter /usr/bin to the program or script. The program is run as soon as possible.

```
CHGCRQA CRQD(CCLIB/CRQRUN) ACTIVITY(RUNACT01)
ACTION(*RUN) GLBNAME(EURO VIRUSCHK EXE 1 US)
CPNAME((EUROITAL FREDSW)) PARM("/usr/bin")
```

- Change an activity to run the program known by the global name EURO WORDPROC EXE 2 US on managed system JOHNSWS. This activity should be run immediately by the PS/2. The object should be kept in the NetView/DM2 catalog. The change control server is in the same network as the local system.

```
CHGCRQA CRQD(CCLIB/CRQRUN) ACTIVITY(RUNACT02)
ACTION(*RUN) GLBNAME(EURO WORDPROC EXE 2 US)
CPNAME((*NETATR JOHNSWS)) KEEPOBJ(*KEEP)
```

### Example 6: Install Actions

The first two examples shown here match those shown for the accept and remove examples.

- The install activity is:

```
ADDCRQA CRQD(CCLIB/CRQACPRMV) ACTIVITY(INSACT01)
ACTION(*INS) GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
CPNAME((ROMSERV1 MARYPWS1)) ACTUSEINS(*PROD)
ALTACTCOMP(*NOTALLOWED) AUTOACP((*YES))
ALWRMV(*YES)
COREQCHGNL((REXX PROC UPDATE CONFIG))
```

Instead of installing the objects on the single system MARYPWS1, the objects should be installed on all the managed systems controlled by the change control server ROMSERV1. (All the managed systems are listed in the node list ROMCLIENTS stored in library ROMLIB.) To install the objects in a removable manner, the install activity is changed to:

```
CHGCRQA CRQD(CCLIB/CRQACPRMV) ACTIVITY(INSACT01)
ACTION(*SAME) GLBNAME(*SAME)
NODL(ROMLIB/ROMCLIENTS) CPNAME(*NONE)
ACTUSEINS(*SAME) ALTACTCOMP(*SAME)
AUTOACP((*NO)) ALWRMV(*DESIRED)
COREQCHGNL(*SAME)
```

- The install activity is:

```
ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT02)
          ACTION(*INS)  GLBNAME(EURO WORDPROD UPD 2 3 US)
          CPNAME((EUROITAL FREDSSWS))  ALWRMV(*YES)
```

Instead of install the objects on the single system MARYPWS1, the objects should be installed on all the managed systems controlled by the change control server EUROITAL. (All the managed systems are listed in the node list EUROCLIENT stored in library EUROLIB.) To install the objects on all the managed systems, the install activity is changed to:

```
CHGCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT02)
          ACTION(*SAME)  GLBNAME(*SAME)
          NODL(EUROLIB/EUROCLIENT)  CPNAME(*NONE)
          ALWRMV(*YES)
```

- Change an activity to install the object identified by the global name IBM 1234567 PMGRAB UPD 1 2 on the change control client machine (managed system) called MARYPWS1. MARYPWS1 is controlled by the change control server ROMSERV1. The object is to be installed in the trial area.

```
CHGCRQA  CRQD(CCLIB/CRQINS)  ACTIVITY(INSACT03)
          ACTION(*INS)  GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          ACTUSEINS(*TRIAL)  ALWRMV(*YES)
```

- Change an activity to install the object identified by the global name IBM 1234567 PMGRAB UPD 1 2 on the change control client machine (managed system) called MARYPWS1. MARYPWS1 is controlled by the change control server ROMSERV1. The object is to be installed permanently in the active area. The activity is to be processed at 8:30 a.m. on 25 December 2002.

```
CHGCRQA  CRQD(CCLIB/CRQINS)  ACTIVITY(INSACT04)
          ACTION(*INS)  GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          ACTUSEINS(*PROD)  ALTACTCOMP(*ALLOWED)
          ALWRMV(*NO)
          RMTSTRTIME((*MGDSYS) (8:30:00 12/25/02))
```

- Change an activity to schedule the install of the objects EURO.WORDPROC.REF.2.US and EURO.WORDPROC.UPD.2.3.US on managed system FREDSSWS. FREDSSWS is controlled by the change control server EUROITAL. The objects are installed to the active area in a removable manner. The installation is not automatically accepted. The installation is scheduled for 3 p.m. on 1 January 2002.

```
CHGCRQA  CRQD(CCLIB/CRQRTVINS)  ACTIVITY(INSACT05)
          ACTION(*INS)  GLBNAME(EURO WORDPROC REF 2 US)
          CPNAME((EUROITAL FREDSSWS))  ALTACTCOMP(*ALLOWED)
          COREQCHGNL(EURO WORDPROC UPD 2 3 US)
          RMTSTRTIME((*MGDSYS) (15:00:00 1/01/02))
          COND((RTVACT01 *EQ *SUCCESS *ALLNODES))
```

The activity can be performed only if the object EURO WORDPROC UPD 2 3 US is successfully retrieved from the change control client BRIGSSWS.

```
ADDCRQA  CRQD(CCLIB/CRQRTVINS)  ACTIVITY(RTVACT01)
          ACTION(*RTV)  GLBNAME(EURO WORDPROC UPD 2 3 US)
          CPNAME((EUROITAL BRIGSSWS))
          STRTIME((22:00:00 12/31/02) (06:00:00 1/01/03))
```

- Change an activity to install the object identified by the global name IBM 1234567 WINDMB UPD 1 2 on the production area of managed system MARYPWS1. If installation fails, the object is removed automatically.

```
CHGCRQA  CRQD(CCLIB/CRQINS)  ACTIVITY(INSACT06)
          ACTION(*INS)  GLBNAME(IBM 1234567 WINDMB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          ACTUSEINS(*PROD)  AUTORMV(*YES)  ALWRMV(*YES)
```

## Example 7: Remove Actions

The examples shown here match the first two examples shown for the install examples and demonstrate how some of the installed objects can be removed.

- The install activity added to the change request description is:

```

ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT01)
          ACTION(*INS)  GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))  ACTUSEINS(*PROD)
          ALTACTCOMP(*NOTALLOWED)  AUTOACP(*YES)
          ALWRMV(*YES)
          COREQCHGNL((REXX PROC UPDATE CONFIG))

```

Instead of installing the objects on the single system MARYPWS1, the objects should be installed on all the managed systems controlled by the change control server ROMSERV1. (All the managed systems are listed in the node list ROMCLIENTS stored in library ROMLIB.) The objects should be installed in a removable manner, if all managed systems support removability. To install the objects in a removable manner, the install activity is changed to:

```

CHGCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT01)
          ACTION(*SAME)  GLBNAME(*SAME)
          NODL(ROMLIB/ROMCLIENTS)  CPNAME(*NONE)
          ACTUSEINS(*SAME)  ALTACTCOMP(*SAME)
          AUTOACP(*NO)  ALWRMV(*DESIRED)
          COREQCHGNL(*SAME)

```

The objects IBM 1234567 PMGRAB UPD 1 2 and REXX PROC UPDATE CONFIG are all installed in a removable manner. You can remove the objects by changing the accept activities to the following:

The activities were added to the change request as follows:

```

ADDCRQA  CRQD(CCLIB/CRQINSACC)  ACTIVITY(RMVACT01)
          ACTION(*RMV)
          GLBNAME(IBM 1234567 PMGRAB UPD 1 2)
          CPNAME((ROMSERV1 MARYPWS1))
          COND((INSACT01 *EQ 20 *SAMENODE))
ADDCRQA  CRQD(CCLIB/CRQINSACC)  ACTIVITY(RMVACT02)
          ACTION(*RMV)  GLBNAME(REXX PROC UPDATE CONFIG)
          CPNAME((ROMSERV1 MARYPWS1))
          COND((INSACT01 *EQ 20 *SAMENODE))

```

But they need to be changed to:

```

CHGCRQA  CRQD(CCLIB/CRQINSACC)  ACTIVITY(RMVACT01)
          ACTION(*SAME)  GLBNAME(*SAME)
          NODL(ROMLIB/ROMCLIENTS)  CPNAME(*NONE)
          COND((INSACT01 *EQ *SUCCESS *ALLNODES))
CHGCRQA  CRQD(CCLIB/CRQINSACC)  ACTIVITY(RMVACT02)
          ACTION(*SAME)  GLBNAME(*SAME)
          NODL(ROMLIB/ROMCLIENTS)  CPNAME(*NONE)
          COND((INSACT01 *EQ *SUCCESS *ALLNODES))

```

- The install activity added to the change request description is:

```

ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT02)
          ACTION(*INS)  GLBNAME(EURO WORDPROD UPD 2 3 US)
          CPNAME((EUROITAL FREDSW))  ALWRMV(*YES)

```

Instead of installing the objects on the single system MARYPWS1, the objects should be installed on all the managed systems controlled by the change control server EUROITAL. (All the managed systems are listed in the node list EUROCLIENT stored in library EUROLIB.) The activity is changed to:

```

CHGCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(INSACT02)
          ACTION(*SAME)  GLBNAME(*SAME)
          NODL(EUROLIB/EUROCLIENT)  CPNAME(*NONE)
          ALWRMV(*YES)

```

To remove the object EURO WORDPROD UPD 2 3 US on all the managed systems, the activity RMVACT01 is added to change request CRQACPRMV:

```

ADDCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(RMVACT01)
          ACTION(*RMV)  GLBNAME(EURO WORDPROD UPD 2 3 US)
          CPNAME((EUROITAL FREDSW))
          COND((INSACT04 *EQ *SUCCESS *SAMENODE))

```

should be changed as follows:

```

CHGCRQA  CRQD(CCLIB/CRQACPRMV)  ACTIVITY(RMVACT01)
          ACTION(*SAME)  GLBNAME(*SAME)
          NODL(EUROLIB/EUROCLIENT)  CPNAME(*NONE)
          COND((INSACT04 *EQ *SUCCESS *ALLNODES))

```



- Change an activity to schedule the removal of the previously installed object EURO.WORDPROD.REF.2.US from managed system FREDSWWS with immediate effect but requiring activation.

```
CHGCRQA  CRQD(CCLIB/CR6)  ACTIVITY(ACT01)  ACTION(*RMV)
          GLBNAME(EURO WORDPROD REF 2 US)
          CPNAME((EUROITAL FREDSWWS))
          ALTACTCOMP(*NOTALLOWED)
```

### Example 8: Retrieve Actions

- Change an activity that causes the file EURO WORDPROC UPD 2 3 US to be retrieved from the managed system JOHNSWS. The file is transferred in a compressed format and then stored in a decompressed format. The SNA compression algorithm is used.

```
CHGCRQA  CRQD(CCLIB/CRQRTV)  ACTIVITY(RTVACT01)
          ACTION(*RTV)  GLBNAME(EURO WORDPROC UPD 2 3 US)
          CPNAME((EUROITAL JOHNSWS))  CPRTYPE(*SNA)
          CPRSTGSTT(*DECOMPRESS)  CPRTFRSTT(*COMPRESS)
          REPLACE(*ALLOWED)
```

- Change an activity that causes the object EURO PCSOFT UPD 2 3 US to be retrieved from the managed system JOHNSWS. The file is to be transferred and stored in a compressed format. The user algorithm NVDMLZW is used. This user algorithm is supported if the object is not decompressed at the central site system.

```
CHGCRQA  CRQD(CCLIB/CRQRTV)  ACTIVITY(RTVACT02)
          ACTION(*RTV)  GLBNAME(EURO PCSOFT UPD 2 3 US)
          CPNAME((EUROITAL JOHNSWS))  CPRTYPE(*USER)
          CPRSTGSTT(*COMPRESS)  CPRTFRSTT(*COMPRESS)
          USRCPRINF(NVDMLZW 37 '/D')  REPLACE(*ALLOWED)
```

- Change an activity to retrieve the most recent nightly sales file from each system. The files are cataloged as EURO SALES system-name date-created. The file is sent and stored in a compressed format. The SNA compression algorithm is used. The file must be retrieved after 10 p.m. on the day the request is submitted but before 6 a.m. the next morning when the stores open. All files retrieved must be added to the distribution repository when they are retrieved.

```
CHGCRQA  CRQD(CCLIB/CRQRTV)  ACTIVITY(RTVACT03)
          ACTION(*RTV)  GLBNAME(EURO SALES *ANY *HIGHEST)
          NODL(STORES)  CPRTYPE(*SNA)  CPRSTGSTT(*COMPRESS)
          CPRTFRSTT(*COMPRESS)  REPLACE(*NO)
          STRTIME((22:00:00 *CURRENT) (06:00:00 *NEXT))
```

### Example 9: Send, Send and Run, and Send and Install Actions

- Change an activity to send the file EURO SPELLCHECK EXE 1 US to the managed system FREDSWWS. FREDSWWS is controlled by the change control server EUROITAL. The file is not compressed. The file should be replaced or added at the managed system.

```
CHGCRQA  CRQD(CCLIB/CRQSND)  ACTIVITY(SNDCT01)
          ACTION(*SND)  GLBNAME(EURO SPELLCHECK EXE 1 US)
          CPNAME((EUROITAL FREDSWWS))  REPLACE(*ALLOWED)
```

- Change an activity to send and install the two objects EURO WORDPROC REF 2 US and EURO WORDPROC UPD 2 3 US on managed system FREDSWWS with the attributes that follow. The installation is done to the active area in a removable manner. The installation is not automatically accepted. It is scheduled for 3 a.m. on 13 April 2002 in the time zone where the managed system is located.

```
CHGCRQA  CRQD(CCLIB/CRQSND)  ACTIVITY(SNDCT02)
          ACTION(*SNDINS)  GLBNAME(EURO WORDPROC REF 2 US)
          CPNAME((EUROITAL FREDSWWS))  ALTACTCOMP(*ALLOWED)
          COREQCHGNL((EURO WORDPROC UPD 2 3 US))
          RMTSTRTIME(*MGDSYS (15:00:00 04/13/02))
```

- Change an activity to send and run the program known by the global name EURO.VIRUSCHK.EXE.1.US on managed system FREDSWWS. Pass the parameter /usr/bin to the program. The program is run as soon as possible.



```
CHGCRQA CRQD(CCLIB/CRQSND) ACTIVITY(SNDCT03)
ACTION(*SNDRUN) GLBNAME(EURO VIRUSCHK EXE 1 US)
CPNAME((EUROITAL FREDSW)) PARM("/usr/bin")
```

- Change activities to retrieve the program identified by the global name CUSTNET PCSOFT WDWAPP VER3 941128 from the PS/2 DEVPS2 controlled by the central site CUSTNET. Send and run the activity on all of the PS/2s in the southeast area. The activity is run at 11 p.m. in the time zone where the PS/2 is located only if the retrieve from the PS/2 DEVPS2 is successful. The activity names are generated.

```
CHGCRQA CRQD(CCLIB/CRQRTVSND) ACTIVITY(*GEN)
ACTION(*RTV)
GLBNAME(CUSTNET PCSOFT WDWAPP VER5 021230)
CPNAME((CUSTNET DEVPS2))
```

```
CHGCRQA CRQD(CCLIB/CRQRTVSND) ACTIVITY(*GEN)
ACTION(*SNDRUN)
GLBNAME(CUSTNET PCSOFT WDWAPP VER5 021230)
NODL(PS2SE)
RMTSTRTIME(*MGDSYS (23:00:00 12/30/02))
COND((*PRV *EQ *SUCCES *SAMENODE))
```

### Example 10: Uninstall Actions

- Change an activity to uninstall the component EURO WORDPROC from managed system FREDSW at the time and date specified. The system must be in a state of inactivity at that time. Do not perform the uninstall action if there are objects waiting to be installed.

```
CHGCRQA CRQD(CCLIB/CRQUNINS) ACTIVITY(UNSACT01)
ACTION(*UNINS)
COMPNAME(EURO WORDPROD) CPNAME((EUROITAL FREDSW))
ALTACTCOMP(*NOTALLOWED) FRCUNINS(*NO) REFLVL("2")
RMTSTRTIME(*MGDSYS (14:00:00 05/20/02))
```

Top

---

## Error messages

None

Top



---

## Change License CRQ Activity (CHGLICCRQA)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Change License CRQ Activity (CHGLICCRQA) command changes a license key distribution activity in a change request description.

A prompt override program will be provided to display the current activity information.

### Restrictions:

1. This command is shipped with public \*EXCLUDE authority.
2. The user must have \*CHANGE authority to the change request description object and \*EXECUTE authority to the library.
3. The user changing the activity does not need to be authorized to any objects that are to be manipulated.
4. If a node list (NODL) value is specified, the node list can only contain entries that have a value of \*SNA for the address type.
5. The license key information does not need to exist in the license repository when a license activity is being added to a change request, but it must exist by the time the activity runs.

### Notes:

1. All conditions must be satisfied before the activity can be performed.
2. The start times indicate when the activity can be started. Actual start times may be later due to network and system delays.
3. All the existing records in the central site license repository containing the license key information for the specified product will be sent to the specified managed system or systems Only those ones that match the system serial number are added to the managed system license repository.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Required, Key, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACTIVITY	Activity	<i>Name, *LAST</i>	Required, Key, Positional 2
ACTION	Action	<i>*SAME, *SND</i>	Optional, Positional 3
PRDID	Product ID	<i>Character value, *SAME, *ALL</i>	Optional, Positional 4
LICTRM	License term	<i>Character value, *SAME, *ALL</i>	Optional, Positional 5

Keyword	Description	Choices	Notes
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: <u>*SAME</u> , *NONE Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
CPNAME	Managed system node names	Single values: <u>*SAME</u> , *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name</i> , *NETATR	
	Element 2: Control point	<i>Communications name</i>	
TEXT	Text 'description'	<i>Character value</i> , <u>*SAME</u> , *GEN, *BLANK	Optional
COND	Activity conditions	Single values: <u>*SAME</u> , *NONE Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Activity	<i>Generic name, name</i> , <u>*PRV</u>	
	Element 2: Relational operator	<u>*EQ</u> , *GT, *LT, *LE, *NE, *GE	
	Element 3: End code	0-99, <u>*SUCCESS</u> , *FAIL, *NOTRUN, *ANY	
	Element 4: Condition mode	<u>*ALLNODES</u> , *SAMENODE	
STRTIME	Activity start time	<i>Element list</i>	Optional
	Element 1: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , <u>*SAME</u> , *CURRENT	
	Element 2: Date	<i>Date</i> , <u>*SAME</u> , *CURRENT, *NEXT	
	Element 2: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , <u>*SAME</u> , *ANY, *CURRENT	
	Element 2: Date	<i>Date</i> , <u>*SAME</u> , *ANY, *CURRENT, *NEXT	
HOLD	Hold activity	<u>*SAME</u> , *NO, *YES	Optional

Top

---

## Change request description (CRQD)

Change request description object name.

The possible library values are:

\*LIBL All of the libraries in the user and system portions of the job's library list are searched.

\*CURLIB

The current library for the job is used to locate the object.

*library-name*

Only the library named in this parameter is searched.

The possible value for the change request description is:

*change-request-description*

The name of the change request description object.

Top

---

## Activity (ACTIVITY)

The name of the activity to change in the change request description.

### **\*LAST**

The activity is the last to run in the change request. Only one activity named \*LAST can exist in the change request description.

### *activity-name*

A 10-character activity name.

Top

---

## Action (ACTION)

The license distribution function that is to be performed.

The possible values are:

### **\*SAME**

The value does not change.

**\*SND** Send the license keys of the specified product and license term to the specified managed systems.

Top

---

## Product ID (PRDID)

Specify the 7-character identifier of the product for which the license key information will be distribution.

The possible values are:

### **\*SAME**

The value does not change.

**\*ALL** The license keys for all the products existing in the central site license repository will be distributed.

### *Product-Identifier*

Specifies the 7-character identifier of the product for which the license key information will be distributed.

Top

---

## License term (LICTRM)

Specifies the license term for which the licenses key information will be sent.

The possible values are:

### **\*SAME**

The value does not change.

**\*ALL** The product licenses for all license term found will be sent.

### *license-term*

Specify the license term in one of the following formats:

**Vx** The authorized usage limit is valid for the entire version of the product or feature.

**VxRy** The authorized usage limit is valid for the entire release of the product or feature.

**VxRyMz**

The authorized usage limit is valid only for a modification of the product.

where x and y can be a number from 0 through 9, and z can be a number 0 through 9 or a letter A through Z.

Top

---

## Managed systems node list (NODL)

Node list object name that contains a list of systems that are the destinations for the request. Cannot be specified if the control point names (CPNAME) parameter is also specified

The possible values are:

**\*SAME**

The value does not change.

**\*NONE**

The systems on which this activity is to be performed are not identified by a node list. Individual control point names must be specified.

The possible library values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

**\*CURLIB**

The current library for the job is used to locate the node list object.

***library-name***

Only the library named in this keyword is searched.

The possible value for the node list name is:

***node-list-name***

Node list object name which contains the list of systems on which the command is to be performed.

Top

---

## Managed system node names (CPNAME)

The APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if the node list (NODL) parameter is specified.

**\*SAME**

The value does not change

**\*NONE**

The systems on which this activity is to be performed are not identified individually. A node list must be specified.

**\*NETATR**

The network ID of the local system will be used. This is useful when the node being specified is in the same network as the local system.

*network-identifier*

The APPN network identifier of the managed system on which the activity is to be performed, it is a character field of length 8.

*control-point-name*

The APPN control point name of the managed system on which the activity is to be performed, it is a character field of length 8.

Top

---

## Text 'description' (TEXT)

Specifies the activity description.

\*SAME

The value does not change.

\*GEN A text description is generated based upon the action chosen.

*text-description*

Specify a 50-character description of the activity.

Top

---

## Activity conditions (COND)

Specifies which conditions must be met before this activity can be performed. Each condition identifies an activity that must be run before this activity and the value the end codes from that activity must have to allow this activity to run. The default condition is that the previous activity (in alphabetical order) must complete successfully before this activity can be run.

The possible single values are:

\*SAME

The value does not change.

\*NONE

There are no conditions for this activity.

The possible conditioning activity values are:

This is the activity that must be run before this activity.

\*PRV This activity is conditioned on the previous activity. Activities are ordered alphabetically by activity name. If the activity being added is the first activity, a previous activity does not exist and any condition with \*PRV is marked as having been met.

*conditioning-activity-name*

Specify the name of the activity that must be run before this activity. The activity name specified in the activity (ACTIVITY) parameter cannot be specified in the conditioning activity name. An activity cannot be conditioned on itself.

*generic\*-conditioning-activity-name*

Specify the generic name of the activities that must be run before this activity.

The possible relational operator values are:

This element is the relational operator to use when comparing the end code from the conditioning activity.

- \*EQ Equal
- \*GT Greater than
- \*LT Less than
- \*NE Not equal
- \*GE Greater than or equal
- \*LE Less than or equal

The possible conditioning code results are:

**\*SUCCESS**

The activity ended successfully (0 <= end code <= 9). This end code can only be specified with relational operator \*EQ or \*NE.

**\*FAIL** The activity failed (10 <= end code <= 89). This end code can only be specified with relational operator \*EQ or \*NE.

**\*NOTRUN**

The activity was never started (90 <= end code <= 99). This end code is only specified with relational operator \*EQ or \*NE.

**\*ANY** The activity ended with any end code. This end code is only specified with relational operator \*EQ.

*end-code*

Specify an integer value (0-99) that indicates the result of an activity (success or failure). The end codes used by the change request manager and recommended for applications are:

- 00** Activity completed successfully.
- 01-09** Activity completed with warning messages.
- 10-29** Activity did not complete successfully.
- 30-39** Activity was canceled by a user before it completed.
  - 30 = Activity ended with \*CNTRLD option
  - 35 = Activity ended with \*IMMED option
  - 39 = Activity ended with \*FRCFAIL option
- 40-49** Activity was not run due to errors detected by the application.
  - 40 = Activity not run for security reasons
- 90-99** Activity was not run because conditions or schedules were not met.
  - 95 = Scheduled start time expired
  - 99 = Conditions cannot be met

The possible conditioning mode values are:

The condition mode indicates which systems the conditioning activity must have completed on or before this activity can be performed.

**\*ALLNODES**

The conditioning activity specified must have completed on all nodes before this activity runs.

**\*SAMENODE**

When the conditioning activity specified completes for a given node, the activity specified on the ACTIVITY parameter may run for that same node even though the conditioning activity specified



may not have completed for all other nodes. In the case where this activity lists a node not in the conditioning activity, this activity may run for that node; the condition is ignored.

The possible value is:

Top

---

## Activity start time (STRTIME)

Specifies the date and time when this activity can start on the central site system. The current date and time values and next date values are determined when the change request is submitted.

The possible start after time values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This activity can start any time on or after the time when the change request was submitted.

***start-after-time***

Specify the time when this activity can start. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This activity can start on or after the date on which the change request is submitted.

**\*NEXT**

The activity can start on any date after the date the change request is submitted.

***start-after-date***

Specify the date after this activity can start. The date must be specified in the job date format.

The possible start before time values are:

This element is ignored if the start before date is \*ANY.

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time on or before the start before date.

**\*CURRENT**

The activity must start before the time when the change request was submitted on the date specified on the start before date element.

***start-before-time***

Specify the time before the activity must be started. If the activity cannot be started before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where h = hours, m = minutes, and s = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start before date values are:

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time after the start after time and the start after date.

**\*CURRENT**

The activity must start on the date the change request is submitted.

**\*NEXT**

The activity must start by the day after the date the change request is submitted.

***start-before-date***

Specify the date before the activity must start. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

Top

---

## Hold activity (HOLD)

Specifies that the activity be held when the change request is submitted.

**\*SAME**

The value does not change.

**\*NO** The activity is not held and will run when all conditions are met.

**\*YES** The activity is held for all nodes when the change request is submitted. It must be released by a user before it will run.

Top

---

## Examples

### Example 1: Changing an activity to send

```
CHGLICCRQA  CRQD(MYLIB/CR1) ACTIVITY(ACT01) ACTION(*SND)
             PRDID(1ACCOUN) LICTRM(V5R1M0)
             CPNAME((*NETATR SYS1))
```

Change an activity to send the licenses for the 1ACCOUN product, with a license term of V5R1M0, to the license repository of the iSeries server SYS1. If 1ACCOUN product is installed on the managed system, the license is also activated.

Top

---

## Error messages

None

Top

---

## Change Object CRQ Activity (CHGOBJCRQA)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Change Object Change Request Activity (CHGOBJCRQA) command changes an object distribution activity in a change request description. The object referred to in the activity can be an OS/400 object identified by an OS/400 object name or a global name, or a non-OS/400 object such as a PS/2 file which is identified by a global name.

The activity can be conditioned so that it only runs after one or more other activities have completed (successfully or unsuccessfully). The activity can also be scheduled to run at a date and time in the future.

### Restrictions:

1. You must have \*CHANGE authority to the change request description and \*EXECUTE authority to the library.
2. An object can be specified using an OS/400 object name or a global name but not both.
3. The global name can be a maximum of 65-*n* characters in length, where *n* is the number of tokens. A maximum of 10 tokens can be specified.
4. Only OS/400 program objects or file members such as CL and REXX can be run.
5. The object to be distributed cannot reside in the QTEMP library.
6. If a node list (NODL) value is specified, the node list can only contain entries that have a value of \*SNA for the address type.

### Notes:

The following notes provide information on how the command works.

1. When you add the activity, you do not need to be authorized to any objects that are to be manipulated. When you submit a change request, you must be authorized to any objects that are manipulated.
2. The save and restore history for the object is not updated when it is sent or retrieved.
3. Active message queues are not saved when libraries (\*LIB) are sent or retrieved.
4. All conditions must be satisfied before the activity can be run.
5. The start times indicate when the activity can be started. Actual start times can be later due to network delays and system delays.
6. If a global name is to be used, the Add Distribution Catalog Entry (ADDDSTCLGE) command can be used to indicate where the object is located or is to be stored.
7. Authorization to the object specified on the activity is not verified until the activity runs.

Top

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Required, Key, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACTIVITY	Activity	<i>Name, *LAST</i>	Required, Key, Positional 2
ACTION	Action	<i>*SND, *RTV, *DLT, *RUN, *SNDRUN, *INS, *SNDINS, *UNINS, *SAME</i>	Optional
OBJ	Object	Single values: <i>*GLOBAL, *COMPNAME, *SAME</i> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Object	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
GLBNAME	Global name	Single values: <i>*SAME</i> Other values: <i>Element list</i>	Optional
	Element 1: Global name token 1	<i>Character value, *NETID, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 2: Global name token 2	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 3: Global name token 3	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 4: Global name token 4	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 5: Global name token 5	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 6: Global name token 6	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 7: Global name token 7	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 8: Global name token 8	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 9: Global name token 9	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
	Element 10: Global name token 10	<i>Character value, *ANY, *HIGHEST, *LOWEST, *NETID, *CPNAME, *SERVER, *TARGET, *MDDATE, *MDTIME</i>	
OBJTYPE	Object type	<i>Character value, *SAME</i>	Optional
MBR	Member	<i>Name, *FIRST, *LAST, *ALL, *SAME</i>	Optional
DATATYPE	Data type	<i>*UNSPEC, *CL, *REXX, *SAME</i>	Optional

Keyword	Description	Choices	Notes
COMPNAME	Component name	Single values: <u>*SAME</u> Other values: <i>Element list</i>	Optional
	Element 1: Component name token 1	<i>Character value, *NETID</i>	
	Element 2: Component name token 2	<i>Character value, *NETID, *CPNAME</i>	
	Element 3: Component name token 3	<i>Character value, *NETID, *CPNAME</i>	
	Element 4: Component name token 4	<i>Character value, *NETID, *CPNAME</i>	
	Element 5: Component name token 5	<i>Character value, *NETID, *CPNAME</i>	
	Element 6: Component name token 6	<i>Character value, *NETID, *CPNAME</i>	
	Element 7: Component name token 7	<i>Character value, *NETID, *CPNAME</i>	
REFLVL	Refresh level	<i>Character value, *SAME, *ALL</i>	Optional
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: <u>*NONE, *SAME</u> Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CPNAME	Managed system node names	Single values: <u>*NONE, *SAME</u> Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Character value</i>	
TGTRLS	Target release	<u>*CURRENT, *PRV, V5R1M0, V5R2M0, V5R3M0, *SAME</u>	Optional
REPLACE	Replace object	<u>*NO, *YES, *SAME</u>	Optional
DTACPR	Data compression	<u>*SNA, *NONE, *SAME</u>	Optional
KEEPCLGE	Keep catalog entry	<u>*YES, *NO, *SAME</u>	Optional
PARM	Parameters	Values (up to 20 repetitions): <i>Not restricted</i>	Optional
RMTSTRTIME	Start time on managed system	<i>Element list</i>	Optional
	Element 1: Time zone	<u>*LCLSYS, *MGDSYS, *SAME</u>	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time, *CURRENT, *SAME</i>	
	Element 2: Date	<i>Date, *CURRENT, *NEXT, *SAME</i>	
TEXT	Text 'description'	<i>Character value, *GEN, *BLANK, *SAME</i>	Optional
COND	Activity conditions	Single values: <u>*NONE, *SAME</u> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Activity	<i>Generic name, name, *PRV</i>	
	Element 2: Relational operator	<u>*EQ, *GT, *LT, *LE, *NE, *GE</u>	
	Element 3: End code	0-99, <u>*SUCCESS, *FAIL, *NOTRUN, *ANY</u>	
	Element 4: Condition mode	<u>*ALLNODES, *SAMENODE</u>	

Keyword	Description	Choices	Notes
STRTIME	Activity start time	<i>Element list</i>	Optional
	Element 1: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time, *CURRENT, *SAME</i>	
	Element 2: Date	<i>Date, *CURRENT, *NEXT, *SAME</i>	
	Element 2: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time, *ANY, *CURRENT, *SAME</i>	
	Element 2: Date	<i>Date, *ANY, *CURRENT, *NEXT, *SAME</i>	
HOLD	Hold activity	<i>*NO, *YES, *SAME</i>	Optional

Top

---

## Change request description (CRQD)

Specifies the change request description object name. CRQD is a required parameter.

The possible library values are the following:

**\*LIBL** All of the libraries in the user and in the system portions of the job's library list are searched.

**\*CURLIB**

The current library for the job used to locate the object.

*library-name*

Specify only the library named in this parameter is searched.

The possible values are:

*change-request-description*

Specify the name of the change request description object.

Top

---

## Activity (ACTIVITY)

Specifies the name of the activity to change in the change request description.

**\*LAST**

The activity is the last to run in the change request. When \*LAST is specified for the activity (ACTIVITY) parameter, the condition (COND) parameter and the start time (STRTIME) parameter cannot be specified. Only one activity named \*LAST can exist in the change request description.

*activity-name*

Specify a 10-character activity name.

Top

---

## Action (ACTION)

Specifies the object distribution functions to be performed.

**\*SAME**

The value does not change.

**\*SND** Sends the specified object to the specified managed system or systems.

- \*RTV** Retrieves the specified object from the specified managed system or systems. To retrieve an object from more than one system, a global name with an \*ANY token is required so that each retrieved object has a unique global name. Global names with unspecified tokens (\*ALL, \*HIGHEST, or \*LOWEST) are stored in the distribution repository when they are retrieved.
- \*DLT** Deletes the specified object on the specified system or systems.
- \*RUN** Runs the specified program on the specified system or systems. OS/400 program objects (\*PGM), REXX programs, or file members, containing a CL input stream or an REXX procedure, can be run on iSeries managed systems.
- \*SNDRUN**  
Sends the specified program and runs it on the specified system or systems. The program that is sent deletes on completion.
- \*INS** Sends the specified program and runs it on the specified managed system or systems. Only installable objects can be installed.
- \*SNDINS**  
Sends the objects, previously packaged for installation, on the specified managed system or systems, and installs them. Only objects identified by global names can be installed.
- \*UNINS**  
Removes the objects installed on the specified managed system or systems. Only objects identified by global names can be deleted.

Top

---

## Object (OBJ)

Specifies the name of the object that is run, sent, retrieved, or deleted. For send and retrieve actions, the object name represents the name of the object on both the central site system and the managed systems.

The possible values are:

**\*SAME**

The value does not change.

**\*GLOBAL**

The object is identified by the global name specified on GLBNAME parameter.

The possible library values are one of the following:

**\*LIBL** All of the libraries in the user and in the system portions of the job's library list are searched.

**\*CURLIB**

The current library for the job is used to locate the object.

***library-name***

Specify that only the library named in this parameter is searched.

The possible object name value is:

***object-name***

Specify the object name. Only characters A through Z, 0 through 9, dollar sign, at sign, and number sign can be used in the object names.

Top

---

## Global name (GLBNAME)

Specifies a global name, which is a series of tokens that uniquely identify an object in an SNA network. The global name represents the name that is used to locate the appropriate catalog entry on both the central site system and the managed systems. The catalog entry specifies the object that is used on that system. For example, if a retrieve action is specified, the global name is used to determine the object that is retrieved on the managed system. Also, the global name shows the location where it is to be stored on the central site system.

Special values in a token position indicate how to search for the object. By specifying \*ANY in a token position, the token is ignored when searching for the correct object. If multiple objects are found matching the tokens specified, an error is returned.

If an object is sent, the global name must have been previously cataloged so that it is associated with a local object name or loaded into the distribution repository. Retrieved objects for which no catalog entry exists are placed in the distribution repository. The GLBNAME parameter is ignored if the object name is not \*GLOBAL.

The GLBNAME parameter is not valid when the ACTION value is \*UNINS. When the OBJ is \*GLOBAL and the global name maps to an installable object, the global name must have the following structure.

ComponentName REF RefreshLevel

In this example,

- Component names are the tokens before the token with the REF value. It is used to distinguish objects of an installable object from those from another. The component can be between 1 and 7 tokens.
- The REF token is required to identify the global name as an installable object. It can only be specified from token 2 through 8 in the global name.
- The refresh level is a token with a numeric value. The refresh level shows the level of the installable object and must follow the token with the REF value.

The possible single value is:

### \*SAME

The value does not change.

### Element 1: Token 1

#### \*NETID

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

#### *global-name-token*

Specify the first token of the global name. The first token is recommended to be the registered enterprise ID or network ID.

### Elements 2-10: Tokens 2-10

\*ANY Any token value matches when searching for the object where the action is performed. This is useful when retrieving objects for which some of the tokens in the global name are not known or vary between systems.

#### \*HIGHEST

The object with the highest token value has the action performed on it. The token must be ordered. This is useful when a token in a global name is used to indicate a different version of the object and you need to manipulate the object with the highest version level.



**\*LOWEST**

The object with the lowest token value has the action performed on it. The token must be ordered. This is useful when a token in a global name is used to indicate a different version of the object and you need to manipulate the object with the lowest version level.

**\*NETID**

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

**\*CPNAME**

The control point name of this system is used. The control point is determined by the current value of the LCLCPNAME network attribute value.

**\*SERVER**

This token is stored within the change request activity with the value &SERVER, and is replaced by the short name of the change control server when the object is distributed.

**\*TARGET**

This token is stored within the change request activity with the value &TARGET, and is replaced by the short name of the target when the object is distributed.

**\*MDDATE**

This token is stored within the change request activity with the value &DATE, and is replaced when distributed by the date the object was last modified.

**\*MDTIME**

This token is stored within the change request activity with the value &TIME, and is replaced when distributed by the time the object was last modified.

**Note:** For more detailed information about the Change Control Server support see the Managed System Services/400 Use book.

***global-name-token***

Specify one of a series of 1 to 16 character tokens that uniquely identify the object on which the action is to be performed. Characters A through Z and 0 through 9 can be used. Other special values (@, #, and \$) can be used for tokens that represent network IDs and system names.

Top

---

## Object type (OBJTYPE)

Specifies the object type. It is ignored if a global name is used.

**\*SAME**

The value does not change.

**\*FILEDATA**

A file member should be transferred without the file attributes. This is used to move files between an iSeries system and non-iSeries systems. The \*FILE object type can be used with iSeries systems to preserve the file attributes.

***object-type***

Specify the OS/400 object type.

Top

---

## Member (MBR)

Specifies the physical file member name. This is ignored unless the object type is \*FILE or \*FILEDATA.

### **\*SAME**

The value does not change.

**\*ALL** The action should be performed on all members within the physical file. The object type must be \*FILE. \*ALL must be used for file types that do not have members such as device files. \*ALL is not allowed when the action is \*SNDRUN or \*RUN, or \*FIRST and \*LAST for the \*RUN action.

### **\*FIRST**

The action should be performed on the first member (by date added) in the physical file. The member name is determined when the activity is run.

### **\*LAST**

The action should be performed on the last member (by date added) in the physical file. The member name is determined when the activity is run.

### ***member-name***

Specify the member name on which the action should be performed.

Top

---

## **Data type (DATATYPE)**

Specifies the data type of the member. It is used to specify the type of source file that is run on the managed system. This parameter is ignored when a file is not being sent or run.

### **\*SAME**

The value does not change.

### **\*UNSPEC**

Unspecified file member type. If the data type cannot be determined at the managed system, or if the name of the file where this member resides is QCLSRC, then the file member is treated as a CL batch input stream. If the source file is named QREXSRC, the file member is treated as a REXX procedure.

**\*CL** The file member contains control language, in other words, a CL batch input stream.

### **\*REXX**

The file member contains a REXX procedure.

Top

---

## **Component name (COMPNAME)**

A component name, which is the set of tokens of a global name previous to the REF token. It is the object on which the \*UNINS action acts on. COMPNAME is only valid when ACTION(\*UNINS) and OBJ(\*COMPNAME) are specified.

The possible single value is:

### **\*SAME**

The value does not change.

### **Element 1: Token 1**

#### **\*NETID**

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

#### ***component-name-token***

One of a series of 1 to 16 character tokens that uniquely identifies the object on which the action

is to be performed. Characters A through Z and 0 through 9 can be used. Other special values (@, #, and \$) can be used for tokens that represent network IDs and system names.

#### Elements 2-7: Tokens 2-7

##### **\*NETID**

The network ID of this system is used. The network ID is determined by the current value of the LCLNETID network attribute value.

##### **\*CPNAME**

The control point name of this system is used. The network ID is determined by the current value of the LCLCPNAME network attribute value.

##### *component-name-token*

One of a series of 1 to 16 character tokens that uniquely identifies the object on which the action is to be performed. Characters A through Z and 0 through 9 can be used. Other special values (@, #, and \$) can be used for tokens that represent network IDs and system names.

Top

---

## Refresh level (RFRLVL)

The refresh level is the level of the installable object that will be uninstalled. REFLVL is only valid when ACTION(\*UNINS) and OBJ(\*COMPNAME) are specified.

##### **\*SAME**

The value does not change.

**\*ALL** All the installable objects with different levels will be uninstalled.

##### *refresh-level*

Specify the level of the installable object to be uninstalled. The level is a numeric value up to 16 characters.

Top

---

## Managed systems node list (NODL)

Specifies the node list parameter is the object name that contains a list of systems which are the destinations for the activity. This parameter cannot be specified if the control point name (CPNAME) parameter is also specified.

The possible values are:

##### **\*SAME**

The value does not change.

##### **\*NONE**

The systems on which this activity is to be performed are not specified by a node list. Individual control point names must be specified.

The possible library values are one of the following:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

##### **\*CURLIB**

The current library for the job is used to locate the node list object.

*library-name.*

Specify the name of the library to be searched.

The possible node list name value is:

*node-list-name*

Specify the node list object name containing the list of systems on which the activity is to be performed.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if a node list is specified.

**\*SAME**

The value does not change.

**\*NONE**

The systems on which this activity is performed are not identified individually. A node list must be specified.

The possible values are:

**\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

*network-identifier*

Specify the APPN network identifier of the managed system on which the activity is to be performed.

The possible control point name value is:

*control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed. For NetView Distribution Management Agents, the control point name is the change control client which supports numeric characters (0-9) in the first position of control point names that are valid in other platforms.

Top

---

## Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object. This parameter is ignored for objects with global names that are in the SystemView distribution repository or for actions other than send or retrieve.

**\*SAME**

The value does not change.

**\*CURRENT**

The object is used on the release of the operating system currently running on your system. If V5R3M0 is running on your system, \*CURRENT means that you intend to use the object on a system with V5R3M0 installed. The object can also be used on a system with any later release of the operating system installed.

**\*PRV** The object is intended for a system which is at the previous release level compared to the local system.

**Note:** Modification levels are not supported.

#### *release-level*

Specify the release level in the VxRxMx format. The object is used on a system with the specified release or with any later release of the operating system installed.

Valid values depend on the current version, release, and modification level, and they change with each new release.

Top

---

## Replace object (REPLACE)

Specifies whether the object should be replaced if it already exists. This parameter is ignored for actions other than for send, send and run, or retrieve.

### \*SAME

The value does not change.

**\*NO** An error is returned if the object already exists.

**\*YES** The object is replaced if it already exists.

Top

---

## Data compression (DTACPR)

Specifies that data be compressed when sending or retrieving. This parameter is ignored for actions other than for \*SND, \*SNDRUN, and \*RTV of \*FILEDATA object types. SNA compression with a prime compression character of blank is performed.

### \*SAME

The value does not change.

### **\*NONE**

The file data is not compressed when sent or when retrieved.

**\*SNA** The file data is compressed when sent or when retrieved.

Objects that are globally named can also have compression information specified when they were added to the distribution catalog (ADDDSTCLGE command).

Top

---

## Keep catalog entry (KEEPCLGE)

Specifies if the catalog entry and associated save file corresponding to the installable object will be kept in the specified system or systems. The KEEPCLGE parameter is only valid when ACTION(\*SNDINS) or ACTION(\*INS) is specified.

### \*SAME

The value does not change.

**\*NO** The catalog entry and associated save file are not kept.

**\*YES** The catalog entry and associated save file are kept.

---

## Parameters (PARM)

Specifies parameters to be passed when starting the program. This is ignored if the action is not \*RUN or \*SNDRUN. Up to 20 parameters can be specified.

### \*SAME

The value does not change.

### *parameter*

Specify a 1 to 253 character parameter. The prompt panel initially allows 50 characters to be entered. By entering an ampersand (&) in position 1, the field expands for larger parameters.

Top

---

## Start time on managed system (RMTSTRTIME)

Specifies the date and time when the activity can begin running on the managed system. The current date and time values and the next date values are determined when the activity begins running at the central site system based on the central site date and time.

The possible time zone of the remote start time values are:

### \*SAME

The value does not change.

### \*LCLSYS

The remote start time is specified in the time zone of the central site system.

### \*MGDSYS

The remote start time is specified in the time zone of the managed system.

The possible start after time values are:

This is the definition of the time after which the activity is to start.

### \*SAME

The value does not change.

### \*CURRENT

This function can start on the managed system at any time on or after the time this activity is started on the central site system on the date specified in element 2.

### *start-after-time*

Specify the time when this function can start on the managed system. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

### \*SAME

The value does not change.

### \*CURRENT

This function starts on the managed system on any date on or after the activity starts on the central site system.

#### **\*NEXT**

This function starts on the managed system on any date after the activity starts on the central site system.

#### ***start-after-date***

Specify the date after the functions start on the managed system. The date must be specified in the job date format.

#### **Notes:**

1. The special values \*CURRENT and \*NEXT are specified for the date and the time if the time zone value \*MGDSYS is specified.
2. This parameter can only be specified if \*RUN or \*SNDRUN actions are specified.

Top

---

## **Text 'description' (TEXT)**

Specifies the activity description.

#### **\*SAME**

The value does not change.

**\*GEN** A description is generated based on the action selected.

#### ***text-description***

Specify a 50-character description of the activity.

Top

---

## **Activity conditions (COND)**

Specifies which conditions must be met before this activity can be performed. Each condition identifies an activity that must be run before this activity and the value the end code from that activity must have to allow this activity to run. The default condition is that the previous activity (in alphabetical order) must complete successfully before this activity can be run.

The possible single values are:

#### **\*SAME**

The value does not change.

#### **\*NONE**

There are no conditions for this activity.

The possible conditioning activity values are:

**\*PRV** This activity is conditioned on the previous activity. Activities are ordered alphabetically by activity name. If the activity being added is the first activity, a previous activity does not exist and any condition with \*PRV is marked as having been met.

#### ***conditioning-activity-name***

Specify the name of the activity which must be run before this activity. The activity name specified in the activity (ACTIVITY) parameter cannot be specified in the conditioning activity name. An activity cannot be conditioned on itself.

#### ***generic\*-activity-name***

Specify the generic name of the activities which must run before this activity.

The possible relational operator values are:

This element is the relational operator to use when comparing the end code from an activity.

- \*EQ** Equal
- \*GT** Greater than
- \*LT** Less than
- \*NE** Not equal
- \*GE** Greater than or equal
- \*LE** Less than or equal

The possible condition code results are:

This element is value compared to the actual end code of the conditioning activity.

**\*SUCCESS**

The activity ended successfully (0 <= end code <= 9). This end code can only be specified with relational operator **\*EQ** or **\*NE**.

**\*FAIL** The activity failed (10 <= end code <= 89). This end code can only be specified with relational operator **\*EQ** or **\*NE**.

**\*NOTRUN**

The activity never started (90 <= end code <= 99). This end code is only specified with relational operator **\*EQ** or **\*NE**.

**\*ANY** The activity ended with any end code. This end code is only specified with relational operator **\*EQ**.

***end-code***

Specify an integer value (0-99) that indicates the result of an activity (success or failure). The end code ranges and descriptions are:

- 00** Activity completed successfully.
- 01-09** Activity completed with warning messages.
- 10-29** Activity did not complete successfully.
- 30-39** Activity was canceled by a user before it completed.
  - 30 = Activity ended with **\*CNTRLD** option
  - 35 = Activity ended with **\*IMMED** option
  - 39 = Activity ended with **\*FRCFAIL** option
- 40-49** Activity was not run due to errors detected by the application.
  - 40 = Activity not run for security reasons
- 90-99** Activity was not run because conditions or schedules were not met.
  - 95 = Scheduled start time expired
  - 99 = Conditions cannot be met

The possible condition mode values are:

This element indicates which systems the conditioning activity must have completed on or before this activity can be performed.

**\*ALLNODES**

The conditioning activity specified must complete on all nodes before this activity runs.



**\*SAMENODE**

When the conditioning activity specified completes for a given node, the activity specified on the ACTIVITY parameter can run for that same node even though the conditioning activity specified cannot have completed for all other nodes. In the case where this activity can run for that node; the condition is ignored.

Top

---

## Activity start time (STRTIME)

Specifies the date and the time when this activity can be started on the central site system. The current date and time values, and the next date values are determined when the change request is submitted.

The possible start after time values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This activity can start any time on or after the time when the change request is submitted.

*start-after-time*

Specify the time when this activity can start. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This activity can start on or after the date on which the change request is submitted.

**\*NEXT**

The activity can start on any date after the date the change request is submitted.

*start-after-date*

Specify the date after this activity can start. The date must be specified in the job date format.

The possible start before time values are:

This element is ignored if the start before date is \*ANY.

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time on or before the start before date.

**\*CURRENT**

The activity must start before the time on which the change request was submitted on the date specified on the start before data element. This value cannot be specified if the start before date is \*CURRENT.

*start-before-time*

Specify the time before which the activity must start. If the activity cannot be started before this time, then it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start before date values are:

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time after the start after time and the start after date.

**\*CURRENT**

The activity must start on the date the change request is submitted.

**\*NEXT**

The activity must start by the day after the date the change request is submitted.

***start-before-date***

Specify the date before the activity must start. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

Top

---

## Hold activity (HOLD)

Specifies that the activity be held when the change request is submitted.

**\*SAME**

The value does not change.

**\*NO** The activity is not held. It runs when all conditions and the start time are met.

**\*YES** The activity is held for all nodes when the change request is submitted. It must be released by you before it runs.

Top

---

## Examples

### Example 1: Retrieving a Job Description

```
CHGOBJCRQA  CRQD(MYLIB/CR1)  ACTIVITY(ACT01)
             ACTION(*RTV)  OBJ(QGPL/QXYZ)
             OBJTYPE(*JOB)  CPNAME((*NETATR SYS1))
```

This command changes an activity to retrieve the QGPL/QXYZ job description from the iSeries server SYS1.

### Example 2: Sending a File to All Systems in the Network

```
CHGOBJCRQA  CRQD(MYLIB/CR2)  ACTIVITY(ACT02)
             ACTION(*SND)  OBJ(ACCTLIB/TAXFILE)
             OBJTYPE(*FILE)  TGTRLS(*PRV)  MBR(DEDUCTIONS)
             STRTIME(23:00:00 9/30/02))
             NODL(MYLIB/ACCTSYS)
```

This command changes an activity to send a tax table to all of the iSeries accounting servers identified in the ACCTSYS node list at 11 p.m. on 30 September 2002. The accounting systems are at the previous release level.

### Example 3: Retrieving a Program

```
CHGOBJCRQA  CRQD(MYLIB/CR3)  ACTIVITY(ACT03)
             ACTION(*RTV)  OBJ(*GLOBAL)
             GLBNAME(CUSTNET PCSOFT WDWAPP VER5 020314)
             CPNAME((CUSTNET DEVPS2))
```

```

CHGOBJCRQA  CRQD(MYLIB/CR3)  ACTIVITY(ACT03)
             ACTION(*SND)  OBJ(*GLOBAL)
             GLBNAME(CUSTNET PCSOFT WDWAPP VER5 020314)
             NODL(MYLIB/PS2SE)

CHGOBJCRQA  CRQD(MYLIB/CR3)  ACTIVITY(ACT03)
             ACTION(*RUN)  OBJ(*GLOBAL)
             GLBNAME(CUSTNET PCSOFT WDWAPP VER5 020314)
             COND(*PRV *EQ *SUCCESS *SAMENODE))
             RMTSTRTIME(*MGDSYS (23:00 10/21/02))
             NODL(MYLIB/PS2SE)

```

This command changes activities to retrieve a program from a PS/2, send it to all of the PS/2s in the PS2SE node list, and run it on the PS/2s at 11 p.m. in the time zone where the PS/2 is located. The program runs at each PS/2 after October 21, 2002.

**Example 4: Retrieving a File after 10 p.m.**

```

CHGOBJCRQA  CRQD(MYLIB/CR4)  ACTIVITY(ACT04)
             ACTION(*RTV)  OBJ(*GLOBAL)
             GLBNAME(CUSTNET SALES *ANY *HIGHEST)
             STRTIME((22:00:00 *CURRENT)
                    (06:00:00 *NEXT))
             NODL(MYLIB/STORES)

```

This command changes an activity to retrieve the most recent nightly sales file from each system identified in the STORES node list. The files are cataloged as CUSTNET SALES system-name date-created. The file must be retrieved after 10 p.m. on the day the request is submitted but before 6 a.m. the next morning when the stores open.

**Example 5: Changing an Activity to Send and Install an Object**

```

CHGOBJCRQA  CRQD(MYLIB/CR1)  ACTIVITY(ACT01)
             ACTION(*SNDINS)
             TEXT('New text for changed CRQ')

```

This command changes an activity from sending the object to sending and installing the object. The text is also changed.

Top

---

## Error messages

None

Top



---

## Change Product CRQ Activity (CHGPRDCRQA)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Change Product CRQ Activity (CHGPRDCRQA) command changes a product distribution activity in a change request description.

### Restrictions:

1. This command is shipped with public \*EXCLUDE authority.
2. You must have \*CHANGE authority to the change request description object and \*EXECUTE authority to the library.
3. If you change the activity, you do not need to be authorized to any objects that are to be manipulated. If you submit the change request, you must be authorized to the change request description.
4. The product must be packaged in a save file and cataloged at the time it runs. You can use the Package Product for Distribution (PKGPRDDST) command to package the product.
5. If a NODL value is specified, the node list can only contain entries that have a value of \*SNA for the address type.

### Notes:

The following notes provide information on how the command works.

1. Authorization to the product specified on the activity is not verified until the activity runs.
2. All conditions must be satisfied before the activity can be performed.
3. The start times indicate when the activity can be started. Actual start times can be later due to network and system delays.
4. Action \*DLTCLGE only deletes the save file containing the licensed program and deletes the distribution catalog entry that maps this save file. It does not delete the product.
5. The save file containing the product is not deleted after an installation. This should be done using a \*DLTCLGE action.
6. If TGTRLS(\*ONLY) is specified, the action parameter must be a value other than \*INS.
7. When it is requested to send the license key of the product, a record with the license key information must exist in the central site license repository by the time this activity runs.
8. All the existing records in the central site license repository containing the license key information for the specified product will be sent to the specified managed system or systems. Only those records that match the system serial number are added to the managed system license repository.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Required, Key, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Keyword	Description	Choices	Notes
ACTIVITY	Activity	Name, *LAST	Required, Key, Positional 2
ACTION	Action	*SND, *RTV, *DLTCLGE, *INS, *SNDINS, *SAME	Optional, Positional 3
PRDID	Product ID	Character value, *SAME	Optional, Positional 4
RLS	Release level	Character value, *ONLY, *SAME	Optional
OPTION	Product option	1-99, *BASE, *SAME	Optional
LODTYPE	Product load type	*ALL, *CODE, *LNG, *SAME	Optional
LODID	Load ID	2900-2999, *ALL, *CODE, *SAME	Optional
TGTRLS	Target release	*CURRENT, *PRV, V5R1M0, V5R2M0, V5R3M0, *ONLY, *SAME	Optional
NODL	Managed systems node list	Element list	Optional
	Element 1: Node list	Single values: *NONE, *SAME Other values: Qualified object name	
	Qualifier 1: Node list	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
CPNAME	Managed system node names	Single values: *NONE, *SAME Other values (up to 50 repetitions): Element list	Optional
	Element 1: Network identifier	Communications name, *NETATR	
	Element 2: Control point	Communications name	
KEEPCLGE	Keep catalog entry	*YES, *NO, *SAME	Optional
SNDLICKEY	Send license key	*YES, *NO, *SAME	Optional
RMTSTRTIME	Start time on managed system	Element list	Optional
	Element 1: Time zone	*LCLSYS, *MGDSYS, *SAME	
	Element 2: Start after	Element list	
	Element 1: Time	Time, *CURRENT, *SAME	
	Element 2: Date	Date, *CURRENT, *NEXT, *SAME	
TEXT	Text 'description'	Character value, *GEN, *BLANK, *SAME	Optional
COND	Activity conditions	Single values: *NONE, *SAME Other values (up to 5 repetitions): Element list	Optional
	Element 1: Activity	Generic name, name, *PRV	
	Element 2: Relational operator	*EQ, *GT, *LT, *LE, *NE, *GE	
	Element 3: End code	0-99, *SUCCESS, *FAIL, *NOTRUN, *ANY	
	Element 4: Condition mode	*ALLNODES, *SAMENODE	
STRTIME	Activity start time	Element list	Optional
	Element 1: Start after	Element list	
	Element 1: Time	Time, *CURRENT, *SAME	
	Element 2: Date	Date, *CURRENT, *NEXT, *SAME	
	Element 2: Start before	Element list	
	Element 1: Time	Time, *ANY, *CURRENT, *SAME	
	Element 2: Date	Date, *ANY, *CURRENT, *NEXT, *SAME	
HOLD	Hold activity	*NO, *YES, *SAME	Optional

Top

---

## Change request description (CRQD)

Specifies the change request description object name.

The possible library values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched.

**\*CURLIB**

The current library for the job is used to locate the object.

*library-name*

Specify that only the library named in this parameter is searched.

*change-request-description*

Specify the name of the change request description object.

Top

---

## Activity (ACTIVITY)

Specifies the name of the activity to change in the change request description.

**\*LAST**

The activity is the last to run in the change request. Only one activity named \*LAST can exist in the change request description. When \*LAST is specified for the activity (ACTIVITY) parameter, the condition (COND) parameter and the start time (STRTIME) parameter cannot be specified. Only one activity named \*LAST can exist in the change request description.

*activity-name*

Specify a 10-character activity name.

Top

---

## Action (ACTION)

Specifies the product distribution function that is to be performed.

**\*SAME**

The value does not change.

**\*SND** Send the product to the specified managed systems.

**\*RTV** Retrieve the product from the specified managed systems.

**\*DLTCLGE**

Delete the catalog entry and associated save file for the product from the specified managed systems.

**\*INS** Install the product on the specified managed systems.

**\*SNDINS**

Send and install the specified product on the specified managed systems.

Top

---

## Product ID (PRDID)

Specifies the 7-character identifier of the product for which the action is performed.

The possible values are:

**\*SAME**

The value does not change.

***product-ID***

Specify the 7-character product ID that is used in the activity.

Top

---

## Release level (RLS)

Specifies which version, release, and modification level of the product is used.

The possible values are:

**\*SAME**

The release does not change.

**\*ONLY**

The release level of the product that is installed on your system.

***release-level***

Specify the release level in the format VxRxMy, where Vx is the version number, and My is the modification number. Valid values for *x* are the numbers 0 through 9. Valid values for *y* are the numbers 0 through 9 and the letters A through Z.

Top

---

## Product option (OPTION)

Specifies which of the optional parts of the product given in the PRDID parameter are used.

The possible values are:

**\*SAME**

The value does not change.

**\*BASE**

Only the base part of the product is used. This is only valid when ACTION is \*SND, \*RTV, \*INS and \*SNDINS.

***product-option-number***

Specify the option number for the product load being used. Valid values range from 1 through 99.

Top

---

## Product load type (LODTYPE)

Specifies the product load objects being used.

The possible values are:



**\*SAME**

The value does not change.

**\*ALL** Code and language objects specified on the LODID parameter are used.

**\*CODE**

The object associated with this product load are used.

**\*LNG** The objects associated with the NLV identified on the LODID parameter are used.

Top

---

## Load ID (LODID)

Specifies the load identifier used.

The possible values are:

**\*SAME**

The value does not change.

**\*ALL** All languages for this product option are used.

**\*CODE**

The code load is used.

***product-load-ID***

Specify the load ID of the product when LODTYPE(\*LNG) or LODTYPE(\*ALL) is used. The load ID must be one of the valid IBM national language versions and be specified in the form 29xx. The value of x can be 0 through 9.

Top

---

## Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the product.

**\*SAME**

The value does not change.

**\*ONLY**

The release is determined by the release of the existing product. This value is not valid if more than one release exists for the same product.

**\*CURRENT**

The product is to be used on the release of the operating system currently running on your system.

**\*PRV** The product is to be used on the previous release with modification level 0 of the operating system.

***release-level***

Specify the release level in the format VxRxMx. The product can be used on a system with the specified release or with any later release of the operating system installed. Valid values depend on the current version, release, and modification level, and can change with each new release.

Top

---

## Managed systems node list (NODL)

Specifies that the node list parameter is the object name that contains a list of systems which are the destinations for the activity. This parameter cannot be specified if control point name (CPNAME) parameter is also specified.

The possible values are:

**\*SAME**

The value does not change.

**\*NONE**

The systems on which this activity is to be performed are not specified by a node list. Individual control point names must be specified.

The possible library values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

**\*CURLIB**

The current library for the job is used to locate the node list object.

***library-name***

Specify the name of the library to be searched.

The possible node list name value is:

***node-list-name***

Specify the node list object name containing the list of systems on which the activity is to be performed.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if a node list is specified.

**\*SAME**

The value does not change.

**\*NONE**

The systems on which this activity is performed are not identified individually. A node list must be specified.

The possible values are:

**\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

***network-identifier***

Specify the APPN network identifier of the managed system on which the activity is to be performed.

The possible value is:

***control-point-name***

Specify the APPN control point name of the managed system on which the activity is to be

performed. For NetView Distribution Management Agents, the control point name is the change control client which supports numeric characters (0-9) in the first position of control point names that are valid in other platforms.

Top

---

## Keep catalog entry (KEEPCLGE)

Specifies if the catalog entry and associated save file corresponding to the product is kept on the specified systems. This is only valid if ACTION(\*INS) or ACTION(\*SNDINS) is specified.

The possible values are:

**\*SAME**

The value does not change.

**\*NO** The catalog entry and associated save file are not kept.

**\*YES** The catalog entry and associated save file are kept.

Top

---

## Send license key (SNDLICENSE)

Specifies if the license key is to be sent with the product.

**\*SAME**

The value does not change.

**\*YES** The license key is sent with the product.

**\*NO** The license key is not sent with the product.

Top

---

## Start time on managed system (RMTSTRTIME)

Specifies the date and time when the activity can begin running on the managed system. The current date and time values and next date values are determined when the activity begins running at the central site system based on the central site date and time.

The possible time zone values are:

**\*SAME**

The value does not change.

**\*LCLSYS**

The remote start time is specified in the time zone of the central site system.

**\*MGDSYS**

The remote start time is specified in the time zone of the managed system.

The start after time values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This function can start on the managed system at any time on or after the time this activity is started on the central site system on the date specified in element 2.

### *start-after-time*

Specify the time when this function can be started on the managed system. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The start after date values are:

#### **\*SAME**

The value does not change.

#### **\*CURRENT**

This function starts on the managed system on any date on or after the activity starts on the central site system.

#### **\*NEXT**

This function starts on the managed system on any date after the activity starts on the central site system.

### *start-after-date*

Specify the date after the functions start on the managed system. The date must be specified in the job date format.

#### **Notes:**

1. The special values \*CURRENT and \*NEXT are specified for the date and time if the time zone value \*MGDSYS is specified.
2. This parameter can only be specified if \*INS or \*SNDINS actions are specified.

Top

---

## **Text 'description' (TEXT)**

Specifies the activity description.

The possible values are:

#### **\*SAME**

The value does not change.

**\*GEN** A text description is generated based on the action chosen.

### *text-description*

Specify a 50-character description of the activity.

Top

---

## **Activity conditions (COND)**

Specifies which conditions must be met before this activity can be performed. Each condition identifies the activity which must run before this activity and the value the end codes from that activity must have to allow this activity to run. The default condition is that the previous activity (in alphabetical order) must complete successfully before this activity can be run.

The possible single values are:

#### **\*SAME**

The value does not change.

**\*NONE**

There are no conditions for this activity.

The possible conditioning activities are:

**\*PRV** This activity is conditioned on the previous activity. Activities are ordered alphabetically by activity name. If the activity being added is the first activity, a previous activity does not exist and any condition with \*PRV is marked as having been met.

***conditioning-activity-name***

Specify the name of the activity which must run before this activity. The activity name specified in the activity (ACTIVITY) parameter cannot be specified in the conditioning activity name. An activity cannot be conditioned on itself.

***generic\*-activity-name***

Specify the generic name of the activities which must run before this activity.

The possible relational operator values are:

This element is the relational operator to use when comparing the end code from the conditioning activity.

**\*EQ** Equal

**\*GT** Greater than

**\*LT** Less than

**\*NE** Not equal

**\*GE** Greater than or equal

**\*LE** Less than or equal

The possible condition code results are:

**\*SUCCESS**

The activity ended successfully (0 <= end code <= 9). This end code can only be specified with relational operator \*EQ or \*NE.

**\*FAIL** The activity failed (10 <= end code <= 89). This end code can only be specified with relational operator \*EQ or \*NE.

**\*NOTRUN**

The activity never started (90 <= end code <= 99). This end code can only be specified with relational operator \*EQ or \*NE.

**\*ANY** The activity ended with any end code. This end code is only be specified with relational operator \*EQ.

***end-code***

Specify an integer value (0-99) that indicates the result of an activity (success or failure). The end code ranges and descriptions are:

**00** Activity completed successfully.

**01-09** Activity completed with warning messages.

**10-29** Activity did not complete successfully.

**30-39** Activity was canceled by a user before it completed.

- 30 = Activity ended with \*CNTRLD option
- 35 = Activity ended with \*IMMED option
- 39 = Activity ended with \*FRCFAIL option

- 40-49** Activity was not run due to errors detected by the application.
  - 40 = Activity not run for security reasons
- 90-99** Activity was not run because conditions or schedules were not met.
  - 95 = Scheduled start time expired
  - 99 = Conditions cannot be met

The possible condition mode values are:

This element indicates which systems the conditioning activity must have completed on or before this activity can be performed.

**\*ALLNODES**

The conditioning activity specified must complete on all nodes before this activity runs.

**\*SAMENODE**

When the conditioning activity specified completes for a given node, the activity specified on the ACTIVITY parameter can run for that same node even though the conditioning activity specified cannot have completed for all other nodes. In the case where this activity can run for that node, the condition is ignored.

Top

## Activity start time (STRTIME)

Specifies the date and time when this activity can be started on the central site system. The current date and time values and next date values are determined when the change request is submitted.

The possible start after time values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This activity can start any time on or after the time when the change request is submitted.

***start-after-time***

Specifies the time when this activity can start. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This activity can start on any date on or after the date on which the change request is submitted.

**\*NEXT**

The activity can start on any date after the date the change request is submitted.

***start-after-date***

Specify the date after this activity can start. The date must be specified in the job date format.

The start before time values are:

This element is ignored if start before date is \*ANY.

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time on or before the start before date.

**\*CURRENT**

The activity must start before the time when the change request was submitted on the date specified on the start before date element. This value cannot be specified if the start before date is \*CURRENT.

***start-before-time***

Specify the time before which the activity must start. If the activity cannot be started before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The start before date values are:

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time after the start after time and the start after date.

**\*CURRENT**

The activity must start on the date the change request is submitted.

**\*NEXT**

The activity must start by the day after the date the change request is submitted.

***start-before-date***

Specify the date before the activity must start. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

Top

---

## Hold activity (HOLD)

Specifies that the activity be held when the change request is submitted.

The possible values are:

**\*SAME**

The value does not change.

**\*NO** The activity is not held. It runs when all conditions and the start time are met.

**\*YES** The activity is held for all nodes when the change request is submitted. It must be released by you before it runs.

Top

---

## Examples

### Example 1: Change an Activity to Send a Product

```
CHGPRDCRQA  CRQD(MYLIB/CRQ1)  ACTIVITY(ACT01)
              ACTION(*SND)  PRDID(1ACCPAY)  RLS(V5R2M0)
              OPTION(1)  LODTYPE(*CODE)  LODID(*CODE)
              SNDLICKY(*YES)  CPNAME(((*NETATR  SYS2))
```

Change an activity to send the accounts payable product to the SYS1 system. Option one with the code parts of the product and its license key will be sent.

### Example 2: Change an Activity to Install an Application

```
CHGPRDCRQA  CRQD(MYLIB/CRQ2)  ACTIVITY(*LAST)
             ACTION(*INS)  PRDID(1SCHEDU)  RLS(V5R2M0)
             OPTION(*BASE)  LODTYPE(*ALL)
             LODID(*ALL)  NODL(MYLIB/ALLSYS)
             RMTSTRTIME(*MGDSYS ('23:00:00' *SAME))
```

Change the last activity of the change request CRQ2, so it installs the scheduler application on all the systems specified in the ALLSYS node list. Both the code and the languages for the application will be installed at 11:00 p.m. in the time zone where the managed systems are located.

### Example 3: Change the Last Activity of the Change Request

```
CHGPRDCRQA  CRQD(MYLIB/CRQ3)  ACTIVITY(*LAST)
             ACTION(*DLTCLGE)  PRDID(1HUMRES)  RLS(V5R2M0)
             OPTION(*BASE)  LODTYPE(*ALL)  LODID(2924)
             CPNAME((*NETATR SYS3))
```

Change the last activity of the change request CRQ3, so it deletes the product packaged for distribution for the human resources application on the SYS3 system.

[Top](#)

---

## Error messages

None

[Top](#)



---

## Change Product Object Desc (CHGPRDOBJD)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Change Product Object Description (CHGPRDOBJD) command is used to update product information in object descriptions.

Top

---

### Parameters

Keyword	Description	Choices	Notes
OBJ	Object name	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Object name	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name</i>	
OBJTYPE	Object type	<i>Character value, *ALL</i>	Required, Positional 2
PRDID	Product ID	<i>Character value</i>	Required, Positional 3
RLS	Release level	<i>Character value</i>	Required, Positional 4
OPTION	Product option	1-99, *BASE	Required, Positional 5
LODID	Load ID	<i>Character value, *CODEDFT</i>	Required, Positional 6

Top

---

### Object name (OBJ)

Specifies the qualified name of the object being changed.

The possible values are:

**\*ALL** All objects in the specified library that are of the types specified on the OBJTYPE parameter are changed.

***generic\*-object-name***

Specify the generic name of the object. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be searched only if \*ALL or \*ALLUSR library values can be specified for the name.

***object-name***

Specify the name of the object to be changed.

The possible library value is:

*library-name*

Specify the name of the library where the object is located.

Top

---

## Object type (OBJTYPE)

Specifies the types of objects being changed.

The possible values are:

**\*ALL** All object types that have the specified object name are changed.

*object-type*

Specify the type of object being changed.

**Note:** Object type \*PRDAVL is not valid.

Top

---

## Product (PRDID)

Specifies the 7-character identifier (ID) for the product to which the objects being changed belong.

Top

---

## Release level (RLS)

Specifies the version, release, and modification level of the product to which the objects being changed belong.

Top

---

## Option (OPTION)

Specifies the product option to which the objects being changed belong.

The possible values are:

**\*BASE**

Specifies that the object is part of the base option of the product.

*product-option-number*

Specify the option number of the product option to which the objects being changed belong.

Valid values range from 1 through 99.

Top

---

## Load ID (LODID)

Specifies the ID of the product load to which the objects being changed belong.

The possible values are:

**\*CODEDFT**

The default code load ID, 5001, is used.

*product-load-ID*

Specify a valid language load ID or a valid code load ID.

Top

---

## Examples

```
CHGPRDOBJD  OBJ(MYLIB/*ALL)  OBJTYPE(*ALL)
              PRDID(9XYZ123)  RLS(V5R2M0)
              OPTION(*BASE)   LODID(*CODEDFT)
```

This command changes the object description of all objects in library MYLIB. These objects are now marked to belong to the code load for the base option of product 9XYZ123 for release V5R2M0.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF358A

Release not valid.

#### SMU1307

Not all object descriptions changed.

#### SMU1312

Objects not found.

Top



---

## Change PTF CRQ Activity (CHGPTFCRQA)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Change Program Temporary Fix Change Request Activity (CHGPTFCRQA) command changes a PTF distribution activity in a change request description.

### Restrictions:

1. You must have \*CHANGE authority to the change request description and \*EXECUTE authority to the library.
2. The PTF must be for an iSeries product package using System Manager licensed program. The PTF must be supported using the Work with Supported Products (WRKSPTPRD) command. Software redesigns can be managed for other types of systems by using globally named objects on the Add Object Change Request Activities (ADDOBJCRQA) command or by using Add Change Management Activity (QNSADDCM) API.
3. A PTF save file must exist and be released in order to be sent to another system.
4. A PTF can only be retrieved from a single managed system.
5. If the destination node does not have the Managed System Services licensed program installed, then only the send action is available. The activity is considered successful when the PTF is sent not when it arrives.
6. Delayed PTFs are applied only during the next IPL. Apply all other PTFs immediately while the activity runs on the system or during the next IPL.
7. PTFs marked as delayed can only be applied permanently if they have been previously applied temporarily. Notice that the IPL would have Notice that an IPL would have been required to apply them since they are delayed.
8. PTFs are always removed temporarily except Vertical Licensed Internal Code (VLIC) PTFs which are removed permanently.
9. If a node list (NODL) value is specified, the node list can only contain entries that have a value of \*SNA for the address type.

The following notes provide information on how the command works.

### Notes:

1. Authorization to the product specified on the activity is not verified until the activity runs.
2. All conditions must be met before the activity runs.
3. The start times indicate when the activity can be started. Actual start times can be later due to network and system delays.

Top

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Required, Key, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACTIVITY	Activity	<i>Name, *LAST</i>	Required, Key, Positional 2
ACTION	Action	<i>*SND, *RTV, *DLT, *APY, *SNDAPY, *RMV, *SAME</i>	Optional, Positional 3
PTFID	PTF identifier	<i>Element list</i>	Optional
	Element 1: PTF	<i>Character value, *ALL, *SAME</i>	
	Element 2: Product	<i>Character value, *ONLY, *SAME</i>	
	Element 3: Release level	<i>Character value, *SAME</i>	
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: <i>*NONE, *SAME</i> Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CPNAME	Managed system node names	Single values: <i>*NONE, *SAME</i> Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Communications name</i>	
PTFPART	PTF parts	<i>*PTF, *CVRLTR, *SAME</i>	Optional
CVRLRLNG	Cover letter language	<i>2900-2999, *SRVRQS, *SAME</i>	Optional
APY	Extent of change	<i>*TEMP, *PERM, *LODONLY, *SAME</i>	Optional
DLYAPY	Delayed PTF apply	<i>*NO, *YES, *SAME</i>	Optional
DLYRMV	Delayed PTF remove	<i>*NO, *YES, *SAME</i>	Optional
RMTSTRTIME	Start time on managed system	<i>Element list</i>	Optional
	Element 1: Time zone	<i>*LCLSYS, *MGDSYS, *SAME</i>	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time, *CURRENT, *SAME</i>	
	Element 2: Date	<i>Date, *CURRENT, *NEXT, *SAME</i>	
TEXT	Text 'description'	<i>Character value, *GEN, *BLANK, *SAME</i>	Optional
COND	Activity conditions	Single values: <i>*NONE, *SAME</i> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Activity	<i>Generic name, name, *PRV</i>	
	Element 2: Relational operator	<i>*EQ, *GT, *LT, *LE, *NE, *GE</i>	
	Element 3: End code	<i>0-99, *SUCCESS, *FAIL, *NOTRUN, *ANY</i>	
	Element 4: Condition mode	<i>*ALLNODES, *SAMENODE</i>	

Keyword	Description	Choices	Notes
STRTIME	Activity start time	<i>Element list</i>	Optional
	Element 1: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time, *CURRENT, *SAME</i>	
	Element 2: Date	<i>Date, *CURRENT, *NEXT, *SAME</i>	
	Element 2: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time, *ANY, *CURRENT, *SAME</i>	
	Element 2: Date	<i>Date, *ANY, *CURRENT, *NEXT, *SAME</i>	
HOLD	Hold activity	<i>*NO, *YES, *SAME</i>	Optional

Top

---

## Change request description (CRQD)

Specifies the change request description object name.

The possible library values are the following:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched.

**\*CURLIB**

The current library for the job is used to locate the object.

*library-name*

Specify only the library named in this parameter is searched.

The possible change request description value is:

*change-request-description*

Specify the name of the change request description object.

Top

---

## Activity (ACTIVITY)

Specifies the activity to change in the change request description.

**\*LAST**

The activity is the last to run in the change request.

*activity-name*

Specify a 10-character activity name.

Top

---

## Action (ACTION)

Specifies the function to be performed on the resource.

The possible values are:

**\*SAME**

The value does not change.

**\*SND** Send the specified PTF to the specified managed system. The PTF is not loaded as part of this activity.

- \***RTV** Retrieve the specified PTF from the specified managed system.
- \***DLT** Deletes the PTF save file and cover letter on the specified managed system.
- \***APY** Apply the PTF on the specified system. All requisite PTFs must be applied. A PTF may set in loaded status ("Not applied"). If the PTF is not previously loaded, then the PTF must be loaded before it is applied. To apply all corequisite PTFs, set the status to "loaded" ("Not applied") status. Request to apply one PTF and all corresponding corequisite PTFs are applied.
- \***SNDAPY**  
Send, load, and apply the PTF on the specified system. All requisite PTFs must have already been applied. If the PTF has corresponding corequisite PTFs, set all PTFs to "loaded" ("Not applied") status to apply all of them.
- \***RMV** Temporarily, remove the PTF on the specified systems.

Top

---

## PTF (PTFID)

Specifies the PTF that is to be distributed.

The possible values are:

\*SAME

The value does not change.

*PTF-identifier*

Specify a 7-character PTF identifier.

- \***ALL** All PTFs for the specified product. It can only be specified when applying, deleting, or removing PTFs. A product must be specified.

The product identifier values are:

This is the product identifier of the product to which the PTF is associated.

\*SAME

The value does not change.

\***ONLY**

The PTF identifier specified is associated with only one supported product.

*product-ID*

Specify the product to which the PTF is associated. The product must be specified when PTF identifiers are not unique across products.

The release level of product values are:

This element shows the release level of the product.

\*SAME

The value does not change.

*release(VxRxMx)*

Specify the release level of the product to which the PTF is associated. The format is VxRxMx. The release must be specified when PTF identifiers are not unique across product releases.

Top



---

## Node list name (NODL)

Specifies the node list parameter is the object name that contains a list of systems which are the destinations for the activity. This parameter cannot be specified if the control point name (CPNAME) parameter is also specified.

The possible values are:

**\*SAME**

The value does not change.

**\*NONE**

The systems on which this activity is to be performed are not specified by a node list. Individual control point names must be specified.

The possible library values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

**\*CURLIB**

The current library for the job is used to locate the node list object.

*library-name*

Specify the name of the library to be searched

The possible node list value is:

*node-list-name*

Specify the node list object name containing the list of systems on which the activity is to be performed.

Top

---

## Change Product CRQ Activity (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if a node list is specified.

**\*SAME**

The value does not change.

**\*NONE**

The systems on which this activity is performed are not identified individually. A node list must be specified.

The possible values are:

**\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

*network-identifier*

Specify the APPN network identifier of the managed system on which the activity is to be performed.

The possible control point name value is:

*control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed.

---

## PTF parts (PTFPART)

Specifies whether the PTFs or cover letters should be sent, retrieved, or deleted.

### \*SAME

The value does not change.

**\*PTF** Only the PTF should be handled.

### **\*CVRLTR**

Only the PTF cover letter is handled. \*CVRLTR is valid only with the \*RTV and \*SND actions.

If the delete action is specified, then both the PTF and the cover letter are always deleted.

Top

---

## Cover letter language (CVLTRLNG)

Specifies the language of the cover letter to be sent with the PTF.

### \*SAME

The value does not change.

### **\*SRVRQS**

The cover letter language specified in the service requester entry for each node determines which cover letter that particular node is sent. If a service requester entry is not present or a language is not specified, the default language 2924 is used. If the language specified is not found, then 2924 is sent except when 2926 is requested in which case 2950 is sent.

### *cover-letter-language-ID*

Specify the 4-character language identifier of the cover letter to be sent or retrieved.

Top

---

## Extent of change (APY)

Specifies the extent of change when the PTF is applied.

### \*SAME

The value does not change.

### **\*TEMP**

The PTF is applied temporarily.

### **\*PERM**

The PTF is applied permanently.

### **\*LODONLY**

The PTF is only loaded. This is useful for PTFs which are part of a corequisite group where all PTFs are applied, removed, or permanently applied as a group and the "loaded" ("Not applied") status is the requisite to perform those functions.

Top

---

## Delayed PTF apply (DLYAPY)

Specifies whether immediate PTFs are applied at the time the activity runs in the specified system, or whether either immediate or delayed PTFs are applied in a delayed manner during the next unattended IPL.

### \*SAME

The value does not change.

**\*NO** Immediate PTF is applied at the time the activity runs. If the PTF is marked delayed, it is not applied until the next unattended IPL.

**\*YES** The PTF that is identified, regardless of whether it is defined as delayed or immediate, is applied during the next unattended IPL.

Top

---

## Delayed PTF remove (DLYRMV)

Specifies how a PTF is removed. Remove a PTF at the time the activity is specified on the system or later during the next IPL.

The possible single value is:

### \*SAME

The value does not change.

**\*NO** Removes a PTF at the time an activity runs.

**\*Yes** Removes during the next next unattended IPL.

Top

---

## Start time on managed system (RMTSTRTIME)

Specifies the date and time when the activity can begin running on the managed system. The current date and time values, and next date values are determined when the activity begins running at the central site system based on the central site date and time.

The possible time zone of the remote start time values are:

### \*SAME

The value does not change.

### **LCLSYS**

The remote start time is specified in the time zone of the central site system.

### **\*MGDSYS**

The remote start time is specified in the time zone of the managed system.

The possible start after time values are:

This is the definition of the time after which the activity is to start.

### \*SAME

The value does not change.

### **\*CURRENT**

This function can start on the managed system at any time on or after the time this activity is started on the central site system on the date specified in element 3.

### *start-after-time*

Specify the time when this function can start on the managed system. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

#### **\*SAME**

The value does not change.

#### **\*CURRENT**

This function starts on the managed system on any date on or after the date this activity starts on the central site system.

#### **\*NEXT**

This function starts on the managed system on any date after the date this activity starts on the central site system.

### *start-after-date*

Specify the date after the functions start on the managed system. The date must be specified in the job date format.

#### **Notes:**

1. The special values \*CURRENT and \*NEXT are specified for the date and the time if the time zone value \*MGDSYS is specified.
2. This parameter can only be specified if \*APY, \*RMV, or \*SNDAPY actions are specified.

Top

---

## **Text 'description' (TEXT)**

Specifies the activity description.

#### **\*SAME**

The value does not change.

**\*GEN** A description is generated based on the action selected.

### *text-description*

Specify a 50-character description of the activity.

Top

---

## **Activity conditions (COND)**

Specifies which conditions must be met before this activity can be performed. Each condition identifies an activity which must run before this activity and the value the end code from that activity which must have to allow this activity to run. The default condition is that the previous activity (in alphabetical order) must complete successfully before this activity can be run.

The possible single values are:

#### **\*SAME**

The value does not change.

#### **\*NONE**

There are no conditions for this activity.

The conditioning activity values are:

The activity which must be run before this activity.

**\*PRV** This activity is conditioned on the previous activity. Activities are ordered alphabetically by activity name. If the activity being added is the first activity, a previous activity does not exist and any condition with \*PRV is marked as having been met.

***conditioning-activity-name***

Specify the name of the activity which must be run before this activity. The activity name specified in the activity (ACTIVITY) parameter cannot be specified in the conditioning activity name. An activity cannot be conditioned on itself.

***generic\*-activity-name***

Specify the generic name of the activities which must be run before this activity.

The relational operator values are:

This element is the relational operator to use when comparing the end code from an activity.

**\*EQ** Equal

**\*GT** Greater than

**\*LT** Less than

**\*NE** Not equal

**\*GE** Greater than or equal

**\*LE** Less than or equal

The condition code values are:

**\*SUCCESS**

The activity ended successfully (0 <= end code <= 9). This end code can only be specified with relational operator \*EQ or \*NE.

**\*FAIL** The activity failed (10 <= end code <= 89). This end code can only be specified with relational operator \*EQ or \*NE.

**\*NOTRUN**

The activity never started (90 <= end code <= 99). This end code is only specified with relational operator \*EQ or \*NE.

**\*ANY** The activity ended with any end code. This end code is only specified with relational operator \*EQ.

***end-code***

Specify an integer value (0-99) that indicates the result of an activity (success or failure). The end code ranges and descriptions are:

**00** Activity completed successfully.

**01-09** Activity completed with warning messages.

**10-29** Activity did not complete successfully.

**30-39** Activity was canceled by a user before it completed.

- 30 = Activity ended with \*CNTRLD option
- 35 = Activity ended with \*IMMED option
- 39 = Activity ended with \*FRCFAIL option

**40-49** Activity was not run due to errors detected by the application.

- 40 = Activity not run for security reasons
- 90-99 Activity was not run because conditions or schedules were not met.
  - 95 = Scheduled start time expired
  - 99 = Conditions cannot be met

The possible condition modes are:

The condition mode which indicates which systems the activity must have completed on or before this activity can be performed.

**\*ALLNODES**

The conditioning activity specified must complete on all nodes before this activity runs.

**\*SAMENODE**

When the conditioning activity specified completes for a given node, the activity specified on the ACTIVITY parameter can run for that same node even though the conditioning activity specified cannot have completed for all other nodes. In the case where this activity can run for that node, the condition is ignored.

Top

## Activity start time (STRTIME)

Specifies the date and time when this activity can be started on the central site system. The current date and time values and the next date values are determined when the change request is submitted.

The possible start after time values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This activity can start any time on or after the time when the change request is submitted.

*start-after-time*

Specify the time when this activity can start. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This activity can start on or after the date the change request is submitted.

**\*NEXT**

The activity can start on any date after the date when the change request is submitted.

*start-after-date*

Specify the date after this activity can start. The date must be specified in the job date format.

The possible start before time values are:

This element is ignored if the start before date is \*ANY.

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time on or before the start before date.

**\*CURRENT**

The activity must start before the time when the change request was submitted on the date specified on the start before date element.

***start-before-time***

Specify the time before which the activity must start. If the activity cannot be started before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start before date values are:

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time after the start after time and the start after date.

**\*CURRENT**

The activity must start on the date the change request is submitted.

**\*NEXT**

The activity must start by the day after the date the change request is submitted.

***start-before-date***

Specify the date before the activity must start. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

Top

---

## Hold activity (HOLD)

Specifies that the activity be held when the change request is submitted.

**\*SAME**

The value does not change.

**\*NO** The activity is not held. It runs when all conditions and the start time are met.

**\*YES** The activity is held for all nodes when the change request is submitted. It must be released by you before it runs.

Top

---

## Examples

### Example 1: Applying a Permanent Change to a PTF

```
CHGPTFCRQA  CRQD(MYLIB/CR1)  ACTIVITY(ACT01)
             ACTION(*APY)  APY(*PERM)
             PTFID(SF12345)  CPNAME((*NETATR SYS1))
```

This command changes an activity to permanently apply PTF SF12345 on system SYS1.

### Example 2: Sending a Change to a PTF

```
CHGPTFCRQA  CRQD(MYLIB/CR2)  ACTIVITY(ACT03)
             ACTION(*SND)  PTFID(SF89345)
             PTFPART(*CVRLTR)  CVRLTRLNG(2950)
             NODL(MYLIB/EUROPE SYS)
```

This command changes an activity to send the PTF cover letter to all systems in Europe that are identified in the EUROPE SYS node list.

### Example 3: Changing Activities to Apply a PTF

```
CHGPTFCRQA  CRQD(MYLIB/CR3)  ACTIVITY(ACT01)
             ACTION(*SND)
             PTFID(SF99911)  CPNAME((NET1 SYSX))

CHGPTFCRQA  CRQD(MYLIB/CR3)  ACTIVITY(ACT02)
             ACTION(*APY)  APY(*TEMP)  PTFID(SF99911)
             RMTSTRTIME((*MGDSYS (02:00:00) (03/30/02))
             CPNAME((NET1 SYSX))
```

This command changes activities to send a PTF and apply it temporarily at 2:00 a.m. on a system encountering a problem.

### Example 4: Changing an Activity to Load a PTF

```
CHGPTFCRQA  CRQD(MYLIB/CR4)  ACTIVITY(ACT01)  ACTION(*APY)
             PTFID(SF89555)  CPNAME(*NETATR SYS4)  APY(*LODONLY)
```

This command changes an activity to load only the PTF SF89555, which is part of a corequisite PTF group, on system SYS4.

### Example 5: Changing an Activity to Send and Permanently Apply a PTF in Delayed Mode

```
CHGPTFCRQA  CRQD(MYLIB/CR5)  ACTIVITY(ACT01)  ACTION(*SNDAPY)
             PTFID(SF91388)
             CPNAME(*NETATR SYS5)  DLYAPY(*YES)
```

This command changes an activity to send and permanently apply PTF SF91388 during the next unattended IPL.

[Top](#)

---

## Error messages

None

[Top](#)



# Change Resource CRQ Activity (CHGRSCCRQA)

Where allowed to run: All environments (\*ALL)  
 Threadsafes: No

Parameters  
 Examples  
 Error messages

The Change Resource Change Request Activity (CHGRSCCRQA) command changes a resource activity in a change request description.

## Restrictions:

1. You must have \*CHANGE authority to the change request description and \*EXECUTE authority to the library.
2. A resource other than \*SYSx is not supported for iSeries managed systems.
3. The keylock switch must be in NORMAL position or the request is rejected.
4. If a NODL value is specified, the node list can only contain entries that have a value of \*SNA for the address type.

## Notes:

The following notes provide information on how the command works.

1. Authorization to the product specified on the activity is not verified until the activity runs.
2. All conditions must be satisfied before the activity can run.
3. The start times indicate when the activity can be started. Actual start times can be later due to network and system delays.

Top

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Optional, Key, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACTIVITY	Activity	<i>Name, *LAST</i>	Optional, Key, Positional 2
ACTION	Action	*RESTART, *SAME	Optional, Positional 3
RSC	Resource	<i>Name, *SYS, *SYSA, *SYSB, *SAME</i>	Optional
NODL	Managed systems node list	Single values: *SAME, *NONE Other values: <i>Element list</i>	Optional
	Element 1: Node list	<i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CPNAME	Managed system node names	Single values: *NONE, *SAME Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Character value</i>	

Keyword	Description	Choices	Notes
<b>OPTION</b>	How to end	*CNTRLD, *IMMED, <u>*SAME</u>	Optional
<b>DELAY</b>	Delay time, if *CNTRLD	0-65535, <u>*SAME</u>	Optional
<b>TEXT</b>	Text 'description'	<i>Character value</i> , *GEN, <u>*SAME</u>	Optional
<b>COND</b>	Activity conditions	Single values: *NONE, <u>*SAME</u> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Activity	<i>Generic name, name</i> , <u>*PRV</u>	
	Element 2: Relational operator	<u>*EQ</u> , *GT, *LT, *LE, *NE, *GE	
	Element 3: End code	0-99, <u>*SUCCESS</u> , *FAIL, *NOTRUN, *ANY	
	Element 4: Condition mode	<u>*ALLNODES</u> , *SAMENODE	
<b>STRTIME</b>	Activity start time	<i>Element list</i>	Optional
	Element 1: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *CURRENT, <u>*SAME</u>	
	Element 2: Date	<i>Date</i> , *CURRENT, *NEXT, <u>*SAME</u>	
	Element 2: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *ANY, *CURRENT, <u>*SAME</u>	
	Element 2: Date	<i>Date</i> , *ANY, *CURRENT, *NEXT, <u>*SAME</u>	
<b>RMTSTRTIME</b>	Start time on managed system	<i>Element list</i>	Optional
	Element 1: Time zone	*LCLSYS, *MGDSYS, <u>*SAME</u>	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *CURRENT, <u>*SAME</u>	
	Element 2: Date	<i>Date</i> , *CURRENT, *NEXT, <u>*SAME</u>	
<b>HOLD</b>	Hold activity	*NO, *YES, <u>*SAME</u>	Optional

Top

---

## Change request description (CRQD)

Specifies the change request description object name.

The possible library values are the following:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched.

**\*CURLIB**

The current library for the job is used to locate the object.

***library-name***

Specify that only the library named in this parameter is searched.

The possible change request description value is:

***change-request-description***

Specify the name of the change request description object.

Top

---

## Activity (ACTIVITY)

Specifies the name of the activity to change in the change request description.

### **\*LAST**

The activity is the last to run in the change request. When \*LAST is specified for the activity (ACTIVITY) parameter, the condition (COND) parameter and the start time (STRTIME) parameter cannot be specified. Only one activity named \*LAST can exist in the change request description.

### ***activity-name***

Specify a 10-character activity name.

Top

---

## **Action (ACTION)**

Specifies the functions to be performed on the resource.

### **\*SAME**

The value does not change.

### **\*RESTART**

Use to restart the specified resource. If the resource specified is \*SYS, \*SYSA, or \*SYSB, the managed system stops and restarts.

Top

---

## **Resource (RSC)**

Specifies the resource name.

### **\*SAME**

The value does not change.

**\*SYS** The action is performed against the entire system. For the restart action, this means that the managed system is powered down and restarted. The system panel determines the IPL source.

### **\*SYSA**

This is the same as \*SYS except the IPL source is the A side.

**\*SYSB** This is the same as \*SYS except the IPL source is the B side.

### ***resource-name***

Specify a 16-character resource name on the managed system. This is not supported by iSeries managed systems.

Top

---

## **Managed systems node list (NODL)**

Specifies that the node list parameter is the object name that contains a list of systems which are the destinations for the activity. This parameter cannot be specified if the control point name (CPNAME) parameter is also specified.

The possible values are:

### **\*SAME**

The value does not change.

### **\*NONE**

The systems on which this activity is to be performed are not specified by a node list. Individual control point names must be specified.

The possible library values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

**\*CURLIB**

The current library for the job is used to locate the node list object.

*library-name*

Specify the name of the library to be searched.

The possible node list name value is:

*node-list-name*

Specify the node list object name containing the list of systems on which the activity is to be performed.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if a node list is specified.

**\*SAME**

The value does not change.

**\*NONE**

The systems on which this activity is performed are not identified individually. A node list must be specified.

**\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

*network-identifier*

Specify the APPN network identifier of the managed system on which the activity is to be performed.

The possible control point name value is:

*control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed. For NetView Distribution Management Agents, the control point name is the change control client which supports numeric characters (0-9) in the first position of control point names that are valid in other platforms.

Top

---

## How to end (OPTION)

Specifies how to end. It specifies whether the system allows the active subsystems to end processing of active jobs in a controlled manner, or whether the system ends the job immediately.

**\*SAME**

The value does not change.

**\*CNTRLD**

The jobs are ended in a controlled manner. If the jobs cannot be ended in a controlled manner during the delay period, they are then ended immediately.

**\*IMMED**

Jobs are ended immediately.

Top

---

**Delay time, if \*CNTRLD (DELAY)**

Specifies the number of seconds the system waits for the jobs to end in a controlled manner.

**\*SAME**

The value does not change.

**3600** The system waits one hour before ending the jobs immediately.

*delay-time*

Specify the delay time in seconds.

Top

---

**Text 'description' (TEXT)**

Specifies the activity description.

**\*SAME**

The value does not change.

**\*GEN** A description is generated based on the action specified.

*text-description*

Specify a 50-character description of the activity.

Top

---

**Activity conditions (COND)**

Specifies which conditions must be met before this activity can be performed. Each condition identifies an activity which must be run before this activity and the value the end code from that activity must have to allow this activity to run. The default condition is that the previous activity (in alphabetical order) must complete successfully before this activity can be run.

The possible single values are:

**\*SAME**

The value does not change.

**\*NONE**

There are no conditions for this activity.

The possible conditioning activity values are:

**\*PRV** This activity is conditioned on the previous activity. Activities are ordered alphabetically by activity name. If the activity being added is the first activity, a previous activity does not exist and any condition with \*PRV is marked as having been met.

*conditioning-activity-name*

Specify the name of the activity which must run before this activity. The activity name specified in the activity (ACTIVITY) parameter cannot be specified in the conditioning activity name. An activity cannot be conditioned on itself.

*generic\*-activity-name*

Specify the generic name of the activities which must run before this activity.

The possible relational operator values are:

This element is the relational operator to use when comparing the end code from a conditioning activity.

- \*EQ** Equal
- \*GT** Greater than
- \*LT** Less than
- \*NE** Not equal
- \*GE** Greater than or equal
- \*LE** Less than or equal

The possible condition code values are:

This element is the value compared to the actual end code of the conditioning activity.

**\*SUCCESS**

The activity ended successfully (0 <= end code <= 9). This end code can only be specified with relational operator **\*EQ** or **\*NE**.

**\*FAIL** The activity failed (10 <= end code <= 89). This end code can only be specified with relational operator **\*EQ** or **\*NE**.

**\*NOTRUN**

The activity never started (90 <= end code <= 99). This end code is only specified with relational operator **\*EQ** or **\*NE**.

**\*ANY** The activity ended with any end code. This end code is only specified with relational operator **\*EQ**.

*end-code*

Specify an integer value (0-99) that indicates the result of an activity (success or failure). The end code ranges and descriptions are:

- 00** Activity completed successfully.
- 01-09** Activity completed with warning messages.
- 10-29** Activity did not complete successfully.
- 30-39** Activity was canceled by a user before it completed.
  - 30 = Activity ended with **\*CNTRLD** option
  - 35 = Activity ended with **\*IMMED** option
  - 39 = Activity ended with **\*FRCFAIL** option
- 40-49** Activity was not run due to errors detected by the application.
  - 40 = Activity not run for security reasons
- 90-99** Activity was not run because conditions or schedules were not met.
  - 95 = Scheduled start time expired
  - 99 = Conditions cannot be met

The possible conditioning mode values are:

This element indicates which systems the conditioning activity must have completed on before this activity can be performed.

**\*ALLNODES**

The conditioning activity specified must have completed for all nodes before this activity runs.

**\*SAMENODE**

When the conditioning activity specified completes on a given node, the activity specified on the ACTIVITY parameter can run for that same node even though the conditioning activity specified cannot have completed for all other nodes. In the case where this activity can run for that node, the condition is ignored.

Top

---

## Activity start time (STRTIME)

Specifies the date and time when this activity can be started on the central site system. The current date and time values and the next date values are determined when the change request is submitted.

The possible start after time values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This activity can start any time on or after the time when the change request is submitted.

*start-after-time*

Specify the time when this activity start. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

**\*SAME**

The value does not change.

**\*CURRENT**

This activity can start on or after the date on which the change request is submitted.

**\*NEXT**

The activity can start on any date after the date the change request is submitted.

*start-after-date*

Specify the date after this activity can start. The date must be specified in the job date format.

The possible start before time values are:

This element is ignored if the start before date is \*ANY.

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time on or before the start before date.

**\*CURRENT**

The activity must start before the time when the change request is submitted on the date specified on the start before data element.

*start-before-time*

Specify the time before which the activity must start. If the activity cannot be started before this

time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator such as a colon (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start before date values are:

**\*SAME**

The value does not change.

**\*ANY** The activity can start at any time after the start after time and the start after date.

**\*CURRENT**

The activity must start on the date the change request is submitted.

**\*NEXT**

The activity must start by the day after the date the change request is submitted.

***start-before-date***

Specify the date before the activity must start. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

Top

---

## Start time on managed system (RMTSTRTIME)

Specifies the date and time when the activity can begin running on the managed system. The current date and time values and the next date values are determined when the activity begins running at the central site system based on the central site date and time.

The possible time zone of the remote start time values are:

**\*SAME**

The value does not change.

**\*LCLSYS**

The remote start time is specified in the time zone of the central site system.

**\*MGDSYS**

The remote start time is specified in the time zone of the managed system.

The possible start after time values are:

This is the definition of the time after which the activity is to start.

**\*SAME**

The value does not change.

**\*CURRENT**

This function can start on the managed system at any time on or after the time this activity is started on the central site system on the date specified in element 3.

***start-after-time***

Specify the time after which this function can start on the managed system. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible values of the start after date are:



### \*SAME

The value does not change.

### **\*CURRENT**

This function starts on the managed system on any date on or after the activity starts on the central site system.

### **\*NEXT**

This function starts on the managed system on any date after the activity starts on the central site system.

### *start-after-date*

Specify the date after the functions start on the managed system. The date must be specified in the job date format.

### **Notes:**

1. The special values \*CURRENT and \*NEXT cannot be specified for the date and the time if the time zone value \*MGDSYS is specified.
2. This parameter can only be specified if \*RUN action is specified.

Top

---

## **Hold activity (HOLD)**

Specifies that the activity be held when the change request is submitted.

### \*SAME

The value does not change.

### **\*NO**

The activity is not held. It runs when all conditions and the start time are met.

### **\*YES**

The activity is held for all nodes when the change request is submitted. It must be released by you before it runs.

Top

---

## **Examples**

### **Example 1: Scheduling an IPL on an iSeries Managed System**

```
CHGRSCCRQA  CRQD(MYLIB/CR1) ACTIVITY(ACT01) ACTION(*RESTART)
              RSC(*SYS) STRTIME((04:00:00 *CURRENT))
              NODL(MYLIB/STORES)
```

This activity IPLs the store systems at 4:00 a.m.

### **Example 2: Performing an IPL on a System**

```
CHGRSCCRQA  CRQD(MYLIB/CR1) ACTIVITY(ACT01) ACTION(*RESTART)
              RSC(*SYSA) CPNAME((*NETATR SYS1))
```

This activity performs an IPL to system SYS1 immediately to the A side.

Top

---

## **Error messages**

None



---

## Change Service Provider Attr (CHGSRVPVDA)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Change Service Provider Attributes (CHGSRVPVDA) command displays and changes the configuration attributes of the service provider.

**Restriction:** This command is shipped with public \*EXCLUDE authority and the QSRV and QSRVBAS user profile have private authority to use the command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
LOGRMTPRB	Log remote problems	<u>*SAME</u> , *NONE, *SRVRQS, *ALERT, *ALL	Optional
SNDPTFAUTO	Send PTFs automatically	<u>*SAME</u> , *NO, *YES	Optional
MSGQ	Message queue	Single values: <u>*SAME</u> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Message queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
MAXDST	Maximum distribution	1-5000, <u>*SAME</u> , *NOMAX	Optional
HLDPTF	Hold PTFs received	<u>*SAME</u> , *NO, *YES	Optional
CVRLTRLNG	Cover letter language	<i>Character value</i> , <u>*SAME</u>	Optional
FWDRQSAUTO	Forward requests automatically	<u>*SAME</u> , *NO, *YES	Optional

Top

---

### Log remote problems (LOGRMTPRB)

Specifies when remote problem log entries are created.

The possible values are:

\*SAME

The value does not change.

\*NONE

Problem log entries are not created when service requests or alerts are received.

\*SRVRQS

A problem log entry is created when a service request is received.

\*ALERT

A problem log entry is created when a remote alert is received.

\*ALL A problem log entry is created when an alert, service request, or both are received from a managed system.

---

## Send PTFs automatically (SNDPTFAUTO)

Specifies that program temporary fixes (PTFs) are sent automatically in response to a PTF order or service request.

The possible values are:

**\*SAME**

The value does not change.

**\*NO** A response to a PTF or service request is not automatically sent.

**\*YES** A response to a PTF or service request is automatically sent.

---

## Message queue (MSGQ)

Specifies the name of the message queue to which messages are sent.

The possible value is:

**\*SAME**

The value does not change.

The name of the message queue can be qualified by one of the following library values:

**\*LIBL** All libraries in the job's library list are searched until the first match is found.

**\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, QGPL is used.

*library-name*

Specify the name of the library to be searched.

The possible value message queue name is:

*message-queue-name*

Specify the name of the message queue being changed.

---

## Maximum distribution (MAXDST)

Specifies the maximum number of kilobytes (KB) that can be used for each Program Temporary Fix (PTF) order or service request that receives a response. You can specify a maximum of 5000 KB.

The possible values are:

**\*SAME**

The number of KB does not change.

**\*NOMAX**

Any number of kilobytes can be transmitted.

### *maximum-number-of-kilobytes*

Specify the maximum number of KB for PTFs to be sent. Valid values range from 1 through 5000.

Top

---

## Hold PTFs received (HLDPTF)

Specifies that PTFs that are received are marked as held or are available for distribution.

The possible values are:

### \*SAME

The value does not change.

**\*NO** PTFs that are received are not held and are available for distribution.

**\*YES** PTFs are held, and are not available for distribution.

Top

---

## Cover letter language (CVRLTRLNG)

Specifies the cover letter language to be used.

The possible value is:

### *cover-letter-language*

Specify the cover letter language to be sent or retrieved with a PTF.

Top

---

## Forward requests automatically (FWDRQSAUTO)

Specifies whether requests should be automatically forwarded to the next service provider.

The possible values are:

### \*SAME

The decision to forward requests automatically does not change.

**\*NO** Requests are not forwarded. They are handled at this service provider.

**\*YES** Requests are forwarded. They are handled at the next service provider.

Top

---

## Examples

### Example 1: Creating Remote Log Entries

```
CHGSRVPVDA LOGRMTPRB(*ALERT)
```

This command creates remote problem log entries when alerts are received.

### Example 2: Directing Messages to Another Queue

```
CHGSRVPVDA MSGQ(USERLIB/MSGQ)
```

This command directs all messages created on the service provider, as a result of receiving alert messages or service requests, to the message queue named MSGQ in the library named USERLIB.

**Example 3: Holding PTFs**

```
CHGSRVPVDA  HLDPTF(*YES)
```

This command specifies that all PTFs received through electronic customer support or from distribution media are to be held. They are not available for distribution. You can specify the RLSPTF command to release a PTF for distribution.

**Example 4: Specifying Maximum Size for Distribution**

```
CHGSRVPVDA  MAXDST(1000)
```

This command specifies that PTFs can have a maximum size of only 1000 bytes for distribution through electronic customer support.

[Top](#)

---

## Error messages

None

[Top](#)

---

# Copy PTF to Save File (CPYPTFSAVF)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Copy Program Temporary Fixes (PTFs) to Save Files (CPYPTFSAVF) command allows you to copy all or selected PTFs for installed and supported products from a device into PTF save files.

## Restrictions:

1. Only PTFs for installed and supported products are copied.

The following notes provide information on how the command works.

## Notes:

1. A SAVF is created in the QGPL library for each PTF that is copied.
2. If the PTF exists, the PTF is not copied.
3. When the PTF SAVF is created in the QGPL library, it is named Qxxxxxxx where xxxxxxx is the PTF identification number or time stamp.
4. Cover-letter files in the QGPL library are named Qxxxxxyy, and placed in file QAPZCOVER, where xxxxxx is the PTF identification number and yy is the language of the PTF. For English cover letters yy is blank.

Top

---

## Parameters

Keyword	Description	Choices	Notes
FROMDEV	From device	<i>Name</i>	Required, Positional 1
LICPGM	Product	<i>Character value, *ALL, *SUPPTD</i>	Optional, Positional 2
SELECT	PTF numbers to select	Single values: <b>*ALL</b> Other values (up to 50 repetitions): <i>Character value</i>	Optional
OMIT	PTF numbers to omit	Values (up to 50 repetitions): <i>Character value</i>	Optional

Top

---

## From device (FROMDEV)

Specifies the tape unit or optical device that contains the Program Temporary Fixes (PTFs) to be copied into save files (SAVF). Specify the name of the tape device or optical device.

Top

---

## Product (LICPGM)

Specifies the names of the licensed programs for which PTFs are to be copied.

The possible values are:

**\*ALL** PTFs for all licensed programs that are installed or supported are copied on to the system.

**\*SUPPTD**

Only PTFs for supported licensed programs are copied.

*licensed-program*

Specify the licensed program for which PTFs are copied.

Top

---

## PTF numbers to select (SELECT)

Specifies which PTFs are to be copied. The OMIT parameter cannot be specified if individual PTF identification numbers are specified on SELECT parameter.

The possible values are:

**\*ALL** All PTFs are copied.

*PTF-number*

Specify the PTF identification number of the PTF to be copied. A maximum of 50 PTF identification numbers can be specified.

Top

---

## PTF numbers to omit (OMIT)

Specifies that all PTFs are copied except for those specified on this parameter. Up to 50 PTF identification numbers can be specified for omission.

Top

---

## Examples

### Example 1: Copying PTFs for a Specified Program

```
CPYPTFSAVF FROMDEV(TAP01) LICPGM(5722SS1)
```

This command copies all PTFs for the licensed program named 5722SS1 from the device named TAP01 into save files on the system.

### Example 2: Copying All PTFs

```
CPYPTFSAVF FROMDEV(TAP01) LICPGM(*ALL)
```

This command copies all PTFs for all supported products from the device named TAP01 to a save file on the system.

### Example 3: Copying PTFs for a Supported Product

```
CPYPTFSAVF FROMDEV(TAP01) LICPGM(*SUPPTD)
```



This command copies PTFs for all supported products from the device named TAP01 to a save file on the system.

**Example 4: Copying a Specific PTF**

```
CPYPTFSAVF FROMDEV(OPT01) LICPGM(5722SS1) SELECT(PTF0001)
```

This command copies only the PTF named PTF0001 for the licensed program named 5722SS1 from the device named OPT01 to a save file on the system.

[Top](#)

---

## Error messages

None

[Top](#)



## Create Product Definition (CRTPRDDFN)

Where allowed to run: All environments (\*ALL)  
 Threadsafes: No

Parameters  
 Examples  
 Error messages

The Create Product Definition (CRTPRDDFN) command creates a product definition object. This object contains common information for the product.

Top

### Parameters

Keyword	Description	Choices	Notes
PRDDFN	Product definition	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Product definition	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *CURLIB</i>	
PRDID	Product ID	<i>Character value, *SER</i>	Required, Positional 2
RLS	Release level	<i>Character value</i>	Required, Positional 3
RGSID	Registration identifier	<i>Element list</i>	Required, Positional 4
	Element 1: Registration type	<i>*PHONE, *CUSTOMER</i>	
	Element 2: Registration value	<i>Character value</i>	
MSGF	Message file	<i>Name</i>	Required, Positional 5
PRDOPT	Product options	Values (up to 100 repetitions): <i>Element list</i>	Required, Positional 6
	Element 1: Product option	<i>1-99, *BASE</i>	
	Element 2: Message identifier	<i>Name</i>	
	Element 3: Allow dynamic naming	<i>*NODYNNAM, *ALWDYNNAM</i>	
	Element 4: Language load ID	Single values: <i>*NONE, *IBMLNG, *BASEOPT</i> Other values (up to 51 repetitions): <i>Character value</i>	
Element 5: Code load ID	<i>Character value, *CODEDFT</i>		
CPYRGTFST	Copyright first year	<i>Character value, *CURRENT, *NONE</i>	Required, Positional 7
CPYRGTCUR	Copyright current year	<i>Character value, *CURRENT, *NONE</i>	Optional
ALWMLTRLS	Allow multiple releases	<i>*NO, *YES</i>	Optional
PRDIDSEFX	Product ID suffix	<i>Character value</i>	Optional
RLSDATE	Release date	<i>Date, *NONE</i>	Optional
TEXT	Text 'description'	<i>Character value, *BLANK</i>	Optional
AUT	Authority	<i>*LIBCRTAUT, *USE, *ALL, *CHANGE, *EXCLUDE</i>	Optional

Top

---

## Product definition (PRDDFN)

Specifies the qualified name of the product definition object being created.

The name of the product definition can be qualified by one of the following library values:

### \*CURLIB

The current library for the job is used to create the product definition. If no current library entry exists in the library list, QGPL is used.

### *library-name*

Specify the name of the library where the product definition is to be created.

Top

---

## Product ID (PRDID)

Specifies the 7-character ID of the product for which a product definition is being created. The product identifier must be in the format *nlxxxxx*, where *n* is any numeric character 0 through 9, *l* is any upper-case letter A through Z, and *x* is any numeric character 0 through 9 or upper-case letter A through Z.

The possible values are:

\***SER** The serial number of the system, together with the product identifier suffix (PRDIDSFY) parameter, is used as the product ID. This does not guarantee a unique product ID.

### *product-ID*

Specify the 7-character ID of the product for which a product definition is being created.

Top

---

## Release level (RLS)

Specifies the version, release, and modification level of the product being created.

### *release-level*

Specify the release level in the format *VxRyMz*, where *x* and *y* are 0 through 9, and *z* is 0 through 9 and A through Z.

Top

---

## Registration identifier (RGSID)

Specifies the registration ID of the organization developing the product.

The possible **Registration Type** values are:

### \*PHONE

The 14-character registration value consists of a country or region code, city code, and telephone number.

### \*CUSTOMER

The registration value consists of a country or region code and an IBM customer number.

The possible **Registration Value** values are:

### *registration-value*

Specify the 14-character value to be used as the registration ID.

Top

---

## Message file (MSGF)

Specifies the name of the message file to contain the messages which describe the product and its options. The message ID for the base option is the message ID for the product. The message IDs are listed in the PRDOPT parameter. This message file must be in the base option of the product.

**Note:** If the base product has language features, the message file must be an object in the language load's object list rather than the code load. This allows for a message file to be created for each language and for the message files to be installed in the correct libraries.

Top

---

## Product options (PRDOPT)

Specifies the options allowed for this product.

A product requires the first option defined to be the **base option**. The option for the base option is \*BASE. Specify additional options by selecting Options 1 through 99. Additional options do not have to be specified in sequential order. Specify up to 100 options (the base option and up to 99 additional options). Product options must be listed in the product definition before they can be packaged using the Package Product Option (PKGPRDOPT) command.

The possible **Product Option** values are:

### **\*BASE**

The base option information is created.

### *product-option*

Specify the option number of the product option to which the information applies. Valid values range from 1 through 99.

The possible **message ID** value is:

### *message-ID*

Specify the message ID of the message that describes the product option.

The possible **Allow Dynamic Naming** values are:

### **\*NODYNNAM**

Libraries and root folders are not dynamically named at installation time. The primary library and primary folder names are used except when a secondary language library name is used.

### **\*ALWDYNNAM**

Libraries and root folders can be dynamically named at installation time.

The possible **Language Load ID** values are:

### **\*NONE**

No language loads are used.

### **\*IBMLNG**

The product option is available in all IBM languages.

#### **\*BASEOPT**

The language IDs specified for the base option are used for the language load IDs.

#### ***language-load-ID***

Specify the languages (29xx) in which the option is available.

The possible **Code Load ID** values are:

#### **\*CODEDFT**

The default code load ID, 5001, is used.

#### ***code-load-ID***

Specify the code load ID for this product option. Valid values range from 5001 through 9999.

Top

---

## **Copyright first year (CPYRGTFST)**

Specifies the first copyright year for the product. The year must be specified as a four-digit number, such as 1990.

The possible values are:

#### **\*CURRENT**

The current year is retrieved from the system.

#### **\*NONE**

No first copyright year is specified. The value is stored as 4 blanks.

#### ***first-copyright-year***

Specify the four-digit year of the first copyright.

Top

---

## **Copyright current year (CPYRGTCUR)**

Specifies the current copyright year for the product.

**Note:** If neither CPYRGTCUR(\*NONE) nor CPYRGTFST(\*NONE) is specified, CPYRGTCUR must be greater than or equal to CPYRGTFST. When both copyright years are specified, the CPYRGTCUR must be greater than or equal to the CPYRGTFST.

The possible values are:

#### **\*CURRENT**

The current year is retrieved from the system.

#### **\*NONE**

No current copyright year is specified.

#### ***current-copyright-year***

Specify the copyright year as a four-digit number, such as 1991.

Top

---

## Allow multiple releases (ALWMLTRLS)

Specifies whether more than one release of the product can exist on the system at the same time.

The possible values are:

**\*NO** Only one release of the product is allowed on a system at a time.

**\*YES** The product can be installed more than once if the release levels are distinct.

Top

---

## Product ID suffix (PRDIDSFX)

Specifies the suffix to be used with the product ID when PRDID(\*SER) is specified. Valid characters are uppercase letters A through Z and numbers 0 through 9.

Top

---

## Release date (RLSDATE)

Specifies the release date of the product.

The possible values are:

**\*NONE**

No release date is associated with the product.

*release-date*

Specify the release date in the format determined by the job description.

Top

---

## Text 'description' (TEXT)

Specifies text that briefly describes the object. More information on this parameter is in "Commonly used parameters" section in the CL topic in the iSeries Information Center (<http://www.ibm.com/eserver/series/infocenter>).

The possible values are:

**\*BLANK**

Text is not specified.

*'description'*

Specify no more than 50 characters of text, enclosed in apostrophes.

Top

---

## Authority (AUT)

Specifies the authority given to users who do not have specific authority to the product definition, who are not on the authorization list, and whose user group has no specific authority to the product definition.

The possible values are:

### **\*LIBCRTAUT**

The public authority for the product definition is taken from the CRTAUT value of the target library (the library that is to contain the product definition). This value is determined when the product definition is created. If the CRTAUT value for the library changes after the product definition is created, the new value does not effect any existing objects.

**\*USE** You can perform basic operations on the product load, such as running a program or reading a file. You cannot change the product load. \*USE authority provides operational authority and read authority to the object, and \*EXCLUDE authority to the library.

**\*ALL** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user cannot transfer ownership of the product definition.

### **\*CHANGE**

The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can run and debug the program, but cannot change it. Change authority provides object operational authority and all data authority.

### **\*EXCLUDE**

The user cannot access the product definition.

Top

---

## **Examples**

```
CRTPRDDFN PRDDFN(TESTLIB/TEST01) PRDID(9XYZ123) RLS(V5R2M0)
          RGSID(*PHONE 1234567) MSGF(TSTMSGF)
          PRDOPT((*BASE MSG0001 *NODYNNAM 2924 *CODEDFT)
          CPYRGTFST(2001) CPYRGTCUR(2002) ALWMLTRLS(*NO)
          RLSDATE(*NONE) TEXT('product TEST01')
          AUT(*LIBCRTAUT)
```

This command creates product definition TEST01 in library TESTLIB. The product ID is 9XYZ123, the release level of the product is V5R2M0, and the registration telephone number is 1234567. The message file TSTMSGF in library TESTLIB will contain the messages to describe the product. The base option is used with a language load ID 2924. The remaining parameters are included to further define the object.

Top

---

## **Error messages**

### **\*ESCAPE Messages**

#### **CPF9899**

Error occurred during processing of command.

Top



# Create Product Load (CRTPRDL0D)

Where allowed to run: All environments (\*ALL)  
 Threadsafes: No

Parameters  
 Examples  
 Error messages

The Create Product Load (CRTPRDL0D) command defines a control object for a product option.

Top

## Parameters

Keyword	Description	Choices	Notes
PRDL0D	Product load	Name, *LNG	Required, Positional 1
PRDID	Product ID	Character value	Required, Positional 2
RLS	Release level	Character value	Required, Positional 3
OPTION	Product option	1-99, *BASE	Required, Positional 4
LODTYPE	Product load type	*CODE, *LNG	Required, Positional 5
LODID	Load ID	Character value, *CODEDFT	Required, Positional 6
RGSID	Registration identifier	Single values: *PRDDFN Other values: <i>Element list</i>	Required, Positional 7
	Element 1: Registration type	*PHONE, *CUSTOMER	
	Element 2: Registration value	Character value	
DVLLIB	Principal development library	Name, *PRDDFN, *CODE	Required, Positional 8
PRILIB	Principal primary library	Name, *DVLLIB, *CODE	Optional
PREOPRPGM	Preoperation exit program	Single values: *NONE Other values (up to 10 repetitions): <i>Name</i>	Optional
PSTOPRPGM	Postoperation exit program	Name, *NONE	Optional
MINTGTRLS	Minimum target release	Character value, *CURRENT, *PRV, *CODE, *BASECODE	Optional
LNLIB	Secondary language library	Name	Optional
ADLLIB	Additional libraries	Single values: *NONE Other values (up to 10 repetitions): <i>Element list</i>	Optional
	Element 1: Development library	Name	
	Element 2: Primary library	Name, *DVLLIB, *CODE	
	Element 3: Preoperation exit program	Single values: *NONE Other values (up to 10 repetitions): <i>Name</i>	
	Element 4: Postoperation exit program	Name, *NONE	

Keyword	Description	Choices	Notes
FLRL	Folder list	Single values: <u>*NONE</u> Other values (up to 100 repetitions): <i>Element list</i>	Optional
	Element 1: Development folder	<i>Character value</i>	
	Element 2: Primary folder	<i>Character value, *DVLFLR</i>	
DIRL	Directory list	Single values: <u>*NONE</u> Other values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Home directory	<i>Path name</i>	
	Element 2: Product directory	Values (up to 300 repetitions): <i>Path name, *HOME</i>	
TEXT	Text 'description'	<i>Character value, *BLANK</i>	Optional
AUT	Authority	<u>*LIBCRTAUT</u> , *USE, *ALL, *CHANGE, *EXCLUDE	Optional

Top

---

## Product load (PRDLOD)

Specifies the name of the product load. The product load object is created in the development library (DVLLIB).

### Note:

The name of the language load object and the code load object must be different to ensure both can reside in the same library. A language load can be installed into the same library as the code load if the language matches the primary language of the system.

The possible values are:

**\*LNG** The name of the load object is the same as the previously created language load object for this product, version, release, modification level, and option.

### *product-load-name*

Specify the name of the product load object.

The first language load created for a product option at a given release must be named. Specify PRDLOD(\*LNG) for all other language loads created to ensure all language loads for the product option have the same name.

Top

---

## Product ID (PRDID)

Specifies the identifier (ID) for the product for which a product load is being created. This value must be 7-characters in length (see PRDID parameter for CRTPRDDFN).

Top

---

## Release level (RLS)

Specifies the version, release, and modification level of the product.

Top

---

## Product option (OPTION)

Specifies the product option for which a product load is being created.

The possible values are:

**\*BASE**

The product option is the base option of the product.

***product-option-number***

Specify the option number for the product load being created. Valid values range from 1 through 99.

Top

---

## Product load type (LODTYPE)

Specifies whether the product load object being created describes a language or a code product load.

The possible values are:

**\*CODE**

The objects associated with this product load are not translated.

**\*LNG** The objects associated with this product load are the translatable objects for the option.

Top

---

## Load ID (LODID)

Specifies the load identifier for the product load being created.

The possible values are:

**\*CODEDFT**

The default code load ID, 5001, is used.

***product-load-ID***

Specify a language load ID (29xx) or a valid code load ID. 5001-9999 are valid code load IDs.

Top

---

## Registration identifier (RGSID)

Specifies the registration ID of the product developer.

The possible values are:

**\*PRDDFN**

The registration ID in the product definition of the load being created is used.

The possible registration type values are:

**\*PHONE**

The value used for the 14-character registration value consists of a country or region code, city code or area code, and telephone number.

#### **\*CUSTOMER**

The registration value as a country or region code with an IBM customer number appended to the end.

The possible registration value is:

#### ***registration-value***

Specify the 14-character value to be used as the registration ID.

Top

---

## **Principal development library (DVLLIB)**

Specifies the name of the principal development library. This is the library into which the product load is created.

The possible values are:

#### **\*PRDDFN**

The name of the library in which the product definition exists is used for the development library name.

#### **\*CODE**

The name of the development library for the code load is used.

#### ***development-library-name***

Specify the principal development library name.

Top

---

## **Principal primary library (PRILIB)**

Specifies the name of the principal primary library for the load being created. This is the default library that will be used when the product is installed.

The possible values are:

#### **\*DVLLIB**

The development library name is used as the primary library name.

#### **\*CODE**

The name of the development library for the code load is used.

#### ***primary-library-name***

Specify the principal primary library name.

Top

---

## **Preoperation exit program (PREOPRPGM)**

Specifies the programs needed in the principal library to perform special setup before the product load is saved, restored, or deleted.

The possible values are:

#### **\*NONE**

No exit program for the principal library is called before the product load is saved, restored, or deleted.

### *preoperation-exit-program*

Specify a maximum of 10 programs to be called. Only the first program in the list is called. All other programs in the list can be called by the first program.

Top

---

## Postoperation exit program (PSTOPRPGM)

Specifies the programs needed in the principal library to perform special setup after the product load is saved, restored, or checked.

The possible values are:

### **\*NONE**

No exit program for the principal library are called after the product load is saved or restored.

### *postoperation-exit-program*

Specify the program that is called after the product load is saved with the Save Licensed Program (SAVLICPGM) command, restored with the Restore Licensed Program (RSTLICPGM) command, or checked with the Check Product Option (CHKPRDOPT) command.

Top

---

## Minimum target release (MINTGTRLS)

Specifies the minimum release of the operating system for which the Save Licensed Program (SAVLICPGM) command can save the object.

The code load must specify the earliest target release for a given option. In addition, the code load for the base option must specify the earliest release for a given product. The minimum target release cannot be earlier than V4R5M0.

The possible values are:

### **\*CURRENT**

Uses the version, release, and modification level of the installed operating system. For example, if V5R3M0 is running on the system, then specifying MINTGTRLS(\*CURRENT) is the same as specifying MINTGTRLS(V5R3M0).

**\*PRV** Uses the previous version, release, and modification level of the installed operating system. The previous release of the operating system uses a modification level 0. For example, if V5R3M0 is running on the system, then \*PRV means V5R2M0.

### **\*CODE**

Uses the minimum target release of the code load for this option. This value is valid only for a load type of \*LNG. The code load for this option must exist on the system for this value to be used.

### **\*BASECODE**

Uses the minimum target release of the code load for the base option. The code load for the base option must exist on the system for this value to be used.

### *minimum-target-release*

Specify the version, release, and modification level of the minimum release of the operating system for which the load will be saved. The format is VxRyMz. Valid values for x, y, and z are 0 through 9.

Top

---

## Secondary language library (LNGLIB)

Specifies the name of the secondary language library for the language load being described.

The load and language objects are installed into this library if the language identifier for this load does not match the system primary language ID and no override library name is specified on the Restore Licensed Program (RSTLICPGM) command.

Top

---

## Additional libraries (ADLLIB)

Specifies additional libraries for the product load.

The possible values are:

### \*NONE

The load has no additional libraries specified.

The possible development library values are:

### *development-library-name*

Specify the name of the additional development library.

The possible primary library values are:

### \*DVLlib

The development library name is used.

### \*CODE

The primary library in the code load corresponding to the immediately preceding development library is used. This value is not valid when LODTYPE (\*CODE) is specified.

### *primary-library-name*

Specify the additional primary library name.

The possible preoperation exit program values are:

### \*NONE

No exit program is called for this additional library before the product load is saved, restored, or deleted.

### *preoperation-exit-program*

Specify a maximum of 10 programs to be called. Only the first program in the list is called. All other programs in the list can be called by the first program.

The possible postoperation exit program values are:

### \*NONE

No program for this additional library is called after the product load is saved, restored, or checked.

### *postoperation-exit-program*

Specify the program that is called after the product load is saved with the Save Licensed Program (SAVLICPGM) command, restored with the Restore Licensed Program (RSTLICPGM) command, or checked with the Check Product Option (CHKPRDOPT) command.

Top

---

## Folder list (FLRL)

Specifies the names of the folders assigned to the product load being described. The documents in the development folders are saved when the product load is saved. When creating a code load, the first folder specified must be a root folder. When creating a language load, the first folder specified must be a sub-folder of a root folder. You cannot specify a folder list if any directory names are specified in the DURL parameter.

The possible values are:

### \*NONE

No folders are specified for this product load.

The possible development folder values are:

### *development-folder-path*

Specify the folder path of a folder which is part of the product load.

The possible primary folder values are:

### \*DVLFLR

The development folder path is used for the primary folder path.

### *primary-folder-name*

Specify the folder path for the primary folder. This is the default name of the folder path when the product option is installed.

Top

---

## Directory list (DIRL)

Specifies the names of the directories assigned to the product loads. You cannot specify a directory list if any folder names are specified.

Naming restrictions for assigned directories include:

- You cannot specify /QSYS.LIB and /QDLS directories
- You must specify unique path names
- Do not begin or end the path name with a forward slash
- Do not use a blank in the directory path name
- Do not use any of the character combinations of "." or ".." in the directory path name.

The possible values are:

### \*NONE

No directories are specified for this product load.

The possible home directory values are:

### *home directory*

Specify the path name of the home directory. Up to 300 home directories may be specified.

The possible product directory values are:

### *product-directory*

Specify the product directory to be associated with the home directory. Up to 300 product directories may be specified.

The special system directories named '/QSYS.LIB' and '/QDLS' are not supported and must not be alone or as a containing directory.

**\*HOME**

Use the home directory name for the product directory.

Refer to the System Manager Use book for more information about assigning names to home and product directories.

Top

---

## Text 'description' (TEXT)

Specifies text that briefly describes the product load.

**\*BLANK**

Text is not specified.

**'description'**

Specify no more than 50 characters of text, enclosed in apostrophes.

Top

---

## Authority (AUT)

Specifies the authority given to you if you do not have specific authority to the product load, if you are not on an authorization list, and if your user group has no specific authority to the product load.

**\*LIBCRTAUT**

The public authority for the product load is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the product load). The public authority is determined when the product load is created. If the CRTAUT value for the library changes after the product load is created, the new value does not effect any existing objects.

**\*USE** You can perform basic operations on the product load, such as running a program or reading a file. You cannot change the product load. \*USE authority provides operational authority and read authority to the object, and \*EXCLUDE authority to the library.

**\*ALL** You can perform all operations except those limited to the owner or controlled by authorization list management authority. You can control the object's existence and specify the security for the object, change the object, and perform basic functions on the object. You can change ownership of the product load.

**\*CHANGE**

You can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

**\*EXCLUDE**

The user cannot access the product load.

Top

---

## Examples

### Example 1: Creating a product load



```

CRTPRDL0D  PRDL0D(MYLOAD)  PRDID(9XYZ123)  RLS(V5R2M0)
OPTION(*BASE)  LODTYPE(*CODE)  LODID(5050)
RGSID(*PHONE 1234567)  DVLLIB(TESTLIB)
PRILIB(TESTLIB)  MINTGTRLS(*CURRENT)
TEXT('Product Load *BASE 9XYZ123')  AUT(*LIBCRTAUT)

```

This command creates product load MYLOAD. The product ID is 9XYZ123, the release level of the product is V5R2M0, the option is \*BASE, the load type is \*CODE, the load ID is 5050, and the registration telephone number is 1234567. The development library is TESTLIB and the principal library is also TESTLIB.

**Example 2: Create a product load with one home directory and five product directories**

```

CRTPRDL0D  PRDL0D(MYLOAD)  PRDID(9XYZ123)  RLS(V5R2M0)
OPTION(*BASE)  LODTYPE(*CODE)  LODID(5050)
RGSID(*PHONE 1234567)  DVLLIB(TESTLIB)
PRILIB(TESTLIB)  MINTGTRLS(*CURRENT)
DIRL('/TESTDIR' ('/DIR001' '/DIR002' '/DIR003'
'/DIR004' '/DIR005'))

```

**Example 3: Create a product load with five home directories with each home directory having one product directory**

```

CRTPRDL0D  PRDL0D(MYLOAD)  PRDID(9XYZ123)  RLS(V5R2M0)
OPTION(*BASE)  LODTYPE(*CODE)  LODID(5050)
RGSID(*PHONE 1234567)  DVLLIB(TESTLIB)
PRILIB(TESTLIB)  MINTGTRLS(*CURRENT)
DIRL('/TESTDIR001' (*HOME))
('/TESTDIR002' (*HOME)) ('/TESTDIR003' (*HOME))
('/TESTDIR004' (*HOME)) ('/TESTDIR005' (*HOME))

```

Top

---

## Error messages

### \*ESCAPE Messages

**CPF0CB1**

Registration identifier not valid.

**CPF0C81**

Product load &6 in library &5 not created.

**CPF0C82**

Error occurred while creating product load &6 in library &5.

**CPF0C84**

Load identifier &4 not valid.

**CPF0C9C**

Secondary language library name required.

**CPF0C9D**

Minimum target release not valid.

**CPF0C55**

Registration ID problem with path.

**CPF0C59**

Directory in use.

**CPF0C5B**

Duplicate primary product directory.

**CPF0C5C**

Specified product directory name not allowed.

**CPF0C5D**

Product directory not allowed.

**CPF0C94**

Object name \*LNG not valid for code load.

**CPF0C95**

\*CODE not valid for library.

**CPF0C96**

Secondary language library not valid.

[Top](#)

# Create Program Temporary Fix (CRTPTF)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Create Program Temporary Fix (CRTPTF) command creates a temporary fix for a product that you have developed. The product must be created and installed before using this command.

Top

## Parameters

Keyword	Description	Choices	Notes
PTF	PTF identifier	<i>Character value</i>	Required, Positional 1
LICPGM	Product	<i>Character value</i>	Required, Positional 2
RLS	Release level	<i>Character value</i>	Required, Positional 3
OPTION	Option	1-99, <b>*BASE</b>	Optional
LODID	Load ID	<i>Character value</i> , <b>*CODEDFT</b>	Optional
PTFOBJ	PTF objects	Single values: <b>*NONE</b> Other values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object name	<i>Name</i>	
	Element 2: Object type	<i>Character value</i>	
OBJLIB	Object libraries	<i>Element list</i>	Optional
	Element 1: Development library	<i>Name</i> , <b>*CURLIB</b>	
	Element 2: Primary library	<i>Name</i> , <b>*PRINCIPAL</b>	
PTFDOC	PTF documents	Single values: <b>*NONE</b> Other values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Document	<i>Character value</i>	
	Element 2: Folder	<i>Character value</i>	
DIROBJ	Objects in a directory	Single values: <b>*NONE</b> Other values (up to 30 repetitions): <i>Element list</i>	Optional
	Element 1: Product Directory	<i>Path name</i>	
	Element 2: Development Directory	<i>Path name</i> , <b>*PRDDIR</b>	
	Element 3: Object	Values (up to 100 repetitions): <i>Character value</i>	

Keyword	Description	Choices	Notes
COVER	Cover letters	Single values: <u>*NONE</u> Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Source cover letter file	<i>Qualified object name</i>	
	Qualifier 1: Source cover letter file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL</i>	
	Element 2: Source cover letter member	<i>Name</i>	
	Element 3: National language version	<i>Character value</i>	
PREREQ	Prerequisite PTFs	Single values: <u>*NONE</u> Other values (up to 300 repetitions): <i>Character value</i>	Optional
COREQ	Corequisite PTFs	Single values: <u>*NONE</u> Other values (up to 300 repetitions): <i>Character value</i>	Optional
EXITPGM	PTF exit programs	Single values: <u>*NONE</u> Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Program	<i>Qualified object name</i>	
	Qualifier 1: Program	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *CURLIB</i>	
	Element 2: Run option	<u>*BOTH, *APPLY, *REMOVE, *PREAPY, *PRERMV, *PREBTH</u>	
	Element 3: Type	<u>*PTE, *OBJLST</u>	
	Element 4: User data	<i>Character value, *NONE</i>	
TGTRLS	Target release	<i>Character value, *CURRENT, *PRV</i>	Optional

Top

---

## PTF identifier (PTF)

Specifies the identifier (ID) of the PTF being created. The PTF ID must be in the valid range for the specified product and release.

Top

---

## Product (LICPGM)

Specifies the product ID for which the PTF is being created.

Top

---

## Release level (RLS)

Specifies the version, release, and modification level of the product the PTF is being created for.

Top

---

## Option (OPTION)

Specifies whether the PTF is for the base product or an option of the base product.

The possible values are:

### \*BASE

The PTF is for the base option.

### *product-option-number*

Specify the product option of the PTF being created.

Top

---

## Load ID (LODID)

Specifies the load ID for the PTF being created.

The possible values are:

### \*CODEDFT

The default code load ID, 5001, is used.

### *load-ID*

Specify a language load ID or a valid code load ID. For a language load, the load ID must be one of the valid IBM national language versions and be specified in the form 29xx. For a code load, the load ID must range from 5001 to 9999.

Top

---

## PTF objects (PTFOBJ)

Specifies the objects to be contained in the PTF. The objects must be listed with their associated object type. They must exist in the development library specified and the primary library specified on the OBJLIB parameter. A maximum of 300 objects can be specified.

The possible values are:

### \*NONE

No objects are specified for the PTF.

The possible PTF object name values are:

### *object-name*

Specify the name of the PTF object.

The possible PTF object type values are:

### *object-type*

Specify the PTF object type.

Top

---

## Object libraries (OBJLIB)

Specifies the development library and primary library for this PTF.

The possible development library values are:

### **\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

### ***development-library***

Specify the library where the PTF objects are located.

The possible primary library values are:

### **\*PRINCIPAL**

The PTF objects are associated with the principal library of the specified product option.

### ***primary-library***

Specify the product library with which the PTF objects are associated.

Top

---

## **PTF documents (PTFDOC)**

Specifies the documents to be contained in the PTF. A maximum of 300 documents can be specified.

The possible values are:

### **\*NONE**

No documents are specified for the PTF.

The possible PTF document name values are:

### ***document-name***

Specify the name of the document.

The possible PTF folder name values are:

### ***folder-name***

Specify the folder in which the PTF document will be installed. The create PTF function will append "/QP" to the folder name specified and save the document from the resulting subfolder.

Top

---

## **Objects in a directory (DIROBJ)**

Specifies the product directory that the PTF is for, the development directory where the PTF is found, and the names of objects in the development directory that are included in the PTF.

Up to 30 repetitions are permitted. A product directory, a development directory, and at least one object name must be specified unless DIROBJ(\*NONE) is specified.

The following restrictions exist when you are assigning directory names:

- You cannot specify /QSYS.LIB and /QDLS directories
- You must specify unique path names
- Do not begin or end the path name with a forward slash
- Do not use a blank in the directory path name
- Do not use any of the character combinations of "." or ".." in the directory path name.

A backslash (\) in the directory name is always converted to a forward slash (/). A trailing slash is always ignored. Imbedded blanks and the following characters are not allowed in a directory name:

- Asterisk (\*)
- Question mark (?)
- Apostrophe (')
- Quotation mark (")

If the leading slash (/) is omitted, one will be assumed, and the inclusion of this leading slash must not cause the directory name to exceed the maximum directory name length of 240 characters. The directory name may indicate a directory that is nested in other directories, for example, /A/B/C. The maximum directory name length includes all slashes and all nested directory names.

The possible single value is:

**\*NONE**

The PTF includes no objects that will be stored in directories when the PTF is applied.

**Element 1: Product Directory Name**

*directory-name*

Specify the directory defined by the product which is the default directory where the objects will reside when the PTF is applied. The original objects are replaced with new objects.

**Element 2: Development Directory Name**

**\*PRDDIR**

The development directory is the same as the product directory.

*directory-name*

Specifies the directory where objects that will be included in the PTF currently reside.

**Element 3: Object Name**

*object-name*

Specifies the list of objects in the development directory that are included in this PTF. Up to 100 object names can be specified for each development directory. If more than 100 objects need to be included for a given product directory, the development directory combination can be specified again using a different set of object names.

A given object name must be specified no more than once.

Objects names are limited to a maximum of 255 characters.

Top

**Cover letters (COVER)**

Specifies the file containing the source text for the PTF cover letter. A member is added to the PTF cover letter file QAPZCOVER in library QGPL. The source text data, PTF language, the data that takes the place of the data, and the requisite information is copied to the member. A maximum of 50 cover letters can be specified.

The possible single value is:

**\*NONE**

No cover letters are specified for this PTF.

The possible value for the cover letter member is:

*cover-letter-source-member*

Specify the name of the file member which contains the cover letter text.

The possible library values are:

**\*LIBL** The library search list is used to locate the file.

**library-name**

Specify the name of the library that locates the file.

The possible value for the cover letter file is:

**file-name**

Specify the name of the file which contains the cover letter text.

The possible cover letter language value is:

**NLV** Specify the national language version (NLV) code for the cover letter source. Each language code can be specified only once.

Top

---

## Prerequisite PTFs (PREREQ)

Specifies a list of PTFs that must be applied before applying the current PTF. If these PTFs are not applied, the current PTF will not function. A maximum of 300 prerequisites and corequisites combined can be specified.

The possible values are:

**\*NONE**

No PTFs are required.

**PTF-ID**

Specify the number of the PTF required by this PTF.

Top

---

## Corequisite PTFs (COREQ)

Specifies a list of accompanying PTFs that are required to enable the new PTF. The current PTF is not applied unless all accompanying PTFs listed are also applied. A maximum of 300 prerequisites and corequisites combined can be specified.

The possible values are:

**\*NONE**

No PTFs are required.

**PTF-ID**

Specify the number of the PTF.

Top

---

## PTF exit programs (EXITPGM)

Specifies the exit programs to be called during the PTF apply or remove operation. A maximum of 50 exit programs can be specified.

The possible single value is:



**\*NONE**

No exit programs are specified for this PTF.

The possible program name values are:

***program***

Specify the name of an exit program for this PTF.

The possible library values are:

**\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

***library-name***

Specify the name of the library to be searched.

The possible run option values are:

**\*BOTH**

The exit program is called during both apply and remove operations.

**\*APPLY**

The exit program is called during the apply operation only.

**\*REMOVE**

The exit program is called during the remove operation only.

**\*PREAPY**

The exit program is called before apply PTF processing and at the end of apply PTF processing.

**\*PRERMV**

The exit program is called before and at the end of remove PTF processing.

**\*PREBTH**

The exit program is called before the PTF is applied or removed and at the end of the apply or remove PTF processing.

The possible type values are:

**\*PTF**

The exit program object is included with the PTF objects. The exit program only exists until the PTF is permanently applied.

**\*OBJLST**

The exit program object is not included with the PTF objects. The exit program must exist in the object list for the product, option, release, and load of the PTF or the code load for the product's base option.

The possible user data values are:

**\*NONE**

No user data is passed to the exit program when it is called.

***user-data***

Specify the user data.

Top

---

## Target release (TGTRLS)

Specifies the earliest release of the operating system on which you intend to load and apply the PTF.

## **\*CURRENT**

The PTF is to be loaded and applied on the release of the operating system currently running on your system. The PTF also can be loaded and applied on a system with any later release of the operating system installed.

**\*PRV** The PTF is to be loaded and applied on the previous release with modification level 0 of the operating system. The PTF can be loaded and applied on a system with any later release of the operating system installed.

### *target-level*

Specify the release of the operating system on which you intend to load and apply the PTF. The release level is specified in the format VxRyMz, where Vx is the version, Ry is the release, and Mz is the modification level. Valid values depend on the current version, release, and modification level, and they change with each new release.

Top

---

## **Examples**

### **Example 1: Creating a PTF**

```
CRTPTF PTF(1X00001) LICPGM(1X12345) RLS(V5R2M0)
        OPTION(*BASE) LODID(*CODEDFT) PTFOBJ((X00PGM01 *PGM))
        OBJLIB(PTFDEVLIB *PRINCIPAL)
```

This command creates PTF 1X00001 for release V5R2M0 of product 1X12345. The PTF is created using program X00PGM01 in library PTFDEVLIB. When applied, this PTF replaces program X00PGM01 in the principal product library of the \*BASE option code load. This PTF is created to be loaded and applied on the current release of the operating system or any later release.

### **Example 2: Creating a PTF with a Document**

```
CRTPTF PTF(1X00002) LICPGM(1X12345) RLS(V5R2M0)
        OPTION(*BASE) LODID(*CODEDFT)
        PTFDOC((X00DOC.001 X00FLR.001))
```

This command creates PTF 1X00002 for release V5R2M0 of product 1X12345. The PTF is created using document X00DOC.001 in folder X00FLR.001/QP. Notice that the document is taken from folder X00FLR.001/QP. Create PTF adds "/QP" to the end of the folder specified when searching for the document to be included. When applied, this PTF replaces document X00DOC.001 in folder X00FLR.001 of the \*BASE option code load.

### **Example 3: Creating a PTF with a Cover Letter**

```
CRTPTF PTF(1X00003) LICPGM(1X12345) RLS(V5R2M0)
        OPTION(*BASE) LODID(*CODEDFT)
        COVER((PTFDEVLIB/PTFCVRLTRF PTF1X00003 2924))
        EXITPGM((EXTPGMLIB/QPZ1X00003 *BOTH *PTF))
```

This command creates PTF 1X00003 for release V5R2M0 of product 1X12345. The cover letter for the PTF is created using source text from file PTFCVRLTRF in library PTFDEVLIB member PTF1X00003. The cover letter is created for National Language Version (NLV) 2924. The \*PTF value specified on the exit program parameter causes program QPZ1X00003 to be saved with the PTF as a temporary object. Temporary objects are deleted when the PTF is permanently applied or removed. Note the temporary object naming convention is used for the exit program. Ensure that any temporary object names used are unique for this product release to avoid over writing other PTFs. The \*BOTH value causes this exit program to be called during both apply and remove processing. Temporary objects are identified by QPZ1 at the beginning of the name.

### **Example 4: Creating a PTF for a Previous Release**

```
CRTPTF  PTF(1X00004) LICPGM(1X12345) RLS(V5R2M0)
        OPTION(*BASE) LODID(*CODEDFT)
        PTFOBJ((X00PGM02 *PGM))
        OBJLIB(PTFDEVLIB *PRINCIPAL) TGTRLS(V5R2M0)
```

This command creates PTF 1X00004 for release V5R2M0 of product 1X12345. The PTF is created using program X00PGM02 in library PTFDEVLIB. When applied, this PTF replaces program X00PGM02 in the principal product library of the \*BASE option code load. This PTF is created to be loaded and applied on release V5R2M0 of the operating system or any later release.

Top

---

## Error messages

### \*ESCAPE Messages

#### **CPF35CC**

Library required for PTF operation already exists.

#### **CPF35DC**

Primary library not found.

#### **CPF35D3**

Cover letter not copied.

#### **CPF3505**

Corequisite PTF &1-&2 &3 contains common objects.

#### **CPF3506**

Length of directory name too long.

#### **CPF3507**

Corequisite PTF &1-&2 &3 not specified.

#### **CPF3509**

Specified corequisite PTF &1-&2 &3 not valid.

#### **CPF351A**

Directory object name not valid.

#### **CPF351B**

Duplicate directory object specified.

#### **CPF357A**

Parameter value not valid.

#### **CPF357B**

Product not found.

#### **CPF357D**

Document or folder name not correct.

#### **CPF3571**

PTF ID &1 not within valid range.

#### **CPF3572**

PTF &2-&1 &3 already exists.

#### **CPF3573**

Resources required for product &1 are not available.

#### **CPF3574**

PTF ID not valid.

**CPF358A**

Release not valid.

**CPF358B**

PTF not created.

**CPF358C**

Create PTF not allowed for product &1.

**CPF359C**

Requisite type not valid.

**CPF3901**

PTF &1-&2 &3 not created.

[Top](#)

# Create PTF Package (CRTPTFPKG)

Where allowed to run: All environments (\*ALL)  
 Threadsafes: No

Parameters  
 Examples  
 Error messages

The Create Program Temporary Fix Package (CRTPTFPKG) command creates a package of program temporary fixes (PTFs).

Top

## Parameters

Keyword	Description	Choices	Notes
OUTFILE	Package output file	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Package output file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTMBR	Package output member	<i>Element list</i>	Optional
	Element 1: Member	<i>Name, *FIRST</i>	
	Element 2: Replace or add records	<i>*REPLACE, *ADD</i>	
SELECT	PTFs to select file	Single values: <i>*SUPPTD</i> Other values: <i>Element list</i>	Optional
	Element 1: File	<i>Qualified object name</i>	
	Qualifier 1: File	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
	Element 2: Member	<i>Name, *FIRST</i>	
OMIT	PTFs to omit file	Single values: <i>*NONE</i> Other values: <i>Element list</i>	Optional
	Element 1: File	<i>Qualified object name</i>	
	Qualifier 1: File	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
	Element 2: Member	<i>Name, *FIRST</i>	
SUPERSEDE	Replace Superseded PTFs	<i>*NO, *YES</i>	Optional
PREREQ	Include prerequisite PTFs	<i>*ALL, *NONE, *SAMEPRD</i>	Optional
DEV	Device	<i>Character value, *NONE</i>	Optional
VOL	Volume identifier	<i>Character value, *MOUNTED</i>	Optional
SEQNBR	Sequence number	1-16777215, <i>*END</i>	Optional
ENDOPT	End of media option	<i>*REWIND, *LEAVE, *UNLOAD</i>	Optional
COVER	Cover letter NLV	<i>Character value, *NONE</i>	Optional

Top

---

## Package output file (OUTFILE)

Specifies the physical database file to which the list of PTFs contained in the package is directed. If the output file does not exist, the system creates it.

The name of the file can be qualified by one of the following library values:

**\*LIBL** All libraries in the job's library list are searched until the first match is found.

**\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

*library-name*

Specify the name of the library to be searched.

The possible file name value is:

*file-name*

Specify the name of the file that receives the list of PTFs contained in the package.

Top

---

## Package output member (OUTMBR)

Specifies the name of the database file member to which the list of PTFs contained in the package is directed.

The possible **Member Name** values are:

**\*FIRST**

The first member in the file receives the list of PTFs contained in the package. If the file does not contain a member, the system creates one.

*member-name*

Specify the database file member which receives the list. If the member does not exist, it is added.

The possible **Add or Replace Records** values are:

**\*REPLACE**

The output data replaces any existing records in the specified file member.

**\*ADD** The output data is added to the end of any existing records in the specified file member.

Top

---

## PTFs to select file (SELECT)

Specifies the database file containing the list from which PTFs are selected.

**Note:**

If a PTF has had another take its place the most current PTF is selected.

The possible values are:

**\*SUPPTD**

The system generates the list of PTFs from which to select. The list contains all PTFs for all the products supported on the system.

### Element 1: File Name

The name of the file can be qualified by one of the following library values:

**\*LIBL** All libraries in the job's library list are searched until the first match is found.

**\*CURLIB**

The current library for the job is used searched. If no library is specified as the current library for the job, the QGPL library is used.

*library-name*

Specify the name of the library to be searched.

*file-name*

Specify the name of the file containing the list from which PTFs are selected.

### Element 2: File Member

**\*FIRST**

The first member in the file contains the list of PTFs from which to select. If the file does not contain a member, it is created.

*member-name*

Specify the database file member containing the list of PTFs from which to select. If the member does not exist, it is created.

Top

---

## PTFs to omit file (OMIT)

Specifies the database file which contains the list of PTFs to omit from the package. The PTFs listed in this file that also exist in the file specified on the SELECT parameter are omitted from the package. Only prerequisites and corequisites that are needed on the system are selected.

The possible values are:

**\*NONE**

All PTFs listed in the file specified on the SELECT parameter are included in the package.

### Element 1: File Name

The name of the file can be qualified by one of the following library values:

**\*LIBL** All libraries in the job's library list are searched until the first match is found.

**\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

*library-name*

Specify the name of the library to be searched.

The possible value for the file name is:

*file-name*

Specify the name of the file containing the list of PTFs to omit.

### Element 2: File Member

### **\*FIRST**

The first member in the file contains the list of PTFs to omit. If the file does not contain a member, one is created.

### ***member-name***

Specify the database file member containing the list of PTFs to omit. If the member does not exist, it is created.

Top

---

## **Replace Superseded PTFs (SUPERSEDE)**

Specifies whether superseding PTFs are packaged in place of superseded PTFs.

The possible values are:

**\*YES** Any superseding PTFs are packaged in place of the superseded PTFs.

**\*NO** Superseding PTFs are not packaged in place of the superseded PTFs.

Top

---

## **Include prerequisite PTFs (PREREQ)**

Specifies whether prerequisite PTFs are included in the package. If prerequisite PTFs are included in the package, then corequisite PTFs also are included.

The possible values are:

**\*ALL** Include all prerequisite and corequisite PTFs in this package.

### **\*NONE**

No prerequisites and corequisites are included in the package.

### **\*SAMEPRD**

Prerequisites and corequisites that are in the same product as the PTFs are included in the package.

Top

---

## **Device (DEV)**

Specifies the name of the device to which the PTF package is saved.

The possible values are:

### **\*NONE**

Only the package output file is generated.

### ***device-name***

Specify the name of the device to which the package is saved.

Top



---

## Volume identifier (VOL)

Specifies the volume identifier.

The possible values are:

### \*MOUNTED

The volume currently placed in the device is used.

*volume-identifier*

Specify the volume ID.

Top

---

## Sequence number (SEQNBR)

Specifies the sequence number of the data file that receives the saved package.

The possible values are:

\*END The system saves the package after the last sequence number on the tape.

*sequence-number*

Specify the sequence number of the file. Valid values range from 1 through 16777215.

Top

---

## End of media option (ENDOPT)

Specifies the operation that is automatically performed on the tape or optical volume after the operation ends.

**Note:** This parameter is only valid if a tape or optical device name is specified on the DEV parameter. For optical devices, \*UNLOAD is the only special value supported, \*REWIND and \*LEAVE will be ignored.

The possible values are:

### \*REWIND

The tape is automatically rewound, but not unloaded, after the operation has ended.

### \*LEAVE

The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

### \*UNLOAD

The tape is automatically rewound and unloaded after the operation ends. Some optical devices will eject the volume after the operation ends.

Top

---

## Cover letter NLV (COVER)

Specifies the language used to select the cover letters for this PTF package.

The possible values are:

### \*NONE

Cover letters are not saved.

**29xx** Specify the national language version (NLV) of cover letters being saved.

Top

---

## Examples

Before you can use the CRTPTFPKG command, an output file must be created on the service requester by running the Display PTF (DSPPTF) command and directing the output to a file. For the following examples, output file MYOUTF is created by running the following command:

```
DSPPTF OUTPUT(MYOUTF) OUTFILE(MYOUTF)
```

### Example 1: Creating a PTF Package to Tape

```
CRTPTFPKG OUTFILE(MYOUTF) DEV(TAP01)
```

This command creates a tape with all PTFs that are currently on the service provider but not on the service requester.

### Example 2: Creating a PTF Package to Optical

```
CRTPTFPKG OUTFILE(MYOUTF) DEV(OPT01)
```

This command creates an optical media with all PTFs that are currently on the service provider but not on the service requester.

Top

---

## Error messages

### \*ESCAPE Messages

#### SMU1423

No products currently supported.

#### SMU1424

Program Temporary Fix (PTF) package not created.

#### SMU1431

No PTFs selected for the package.

Top

---

## Delete Product Definition (DLTPRDDFN)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Delete Product Definition (DLTPRDDFN) command allows the user to delete a product definition object.

**Note:** A user who has authority to this command can delete any \*PRDDFN object regardless of whether you have authority to the \*PRDDFN object.

Top

---

### Parameters

Keyword	Description	Choices	Notes
PRDDFN	Product definition, or	<i>Qualified object name</i>	Optional, Positional 1
	Qualifier 1: Product definition, or	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *CURLIB</i>	
PRDID	Product ID	<i>Character value</i>	Optional
RLS	Release level	<i>Character value, <u>*ONLY</u></i>	Optional

Top

---

### Product definition, or (PRDDFN)

Specifies the name of the product definition to be deleted.

**Note:**

You must specify either this parameter or both PRDID and RLS, but all parameters cannot be specified at the same time.

The possible library values are:

**\*CURLIB**

The current library for the job is used to locate the product definition.

***library-name***

Specify the name of the library where the product definition is located.

The possible values are:

***product-definition-name***

Specify the name of the product definition being deleted.

Top

---

## Product (PRDID)

Specifies the 7-character product identifier (ID) of the product for which a definition is being deleted.

Top

---

## Release level (RLS)

Specifies the version, release, and modification level of the product definition to be deleted.

The possible values are:

### \*ONLY

The release level is determined by searching the system for product information for the product specified on the PRDID parameter. This value is not valid if more than one release of the product exists for the same product ID.

### *version-release-modification*

Specify the version, release, and modification level of the product definition being deleted.

Top

---

## Examples

```
DLTPRDDFN PRDDFN(MYLIB/MYPROD)
```

This command deletes the product definition MYPROD located in library MYLIB.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF358A

Release not valid.

Top

---

## Delete Product Load (DLTPRDLOD)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Delete Product Load (DLTPRDLOD) command allows the user to delete a product load object.

### Note:

If you have the authority to use this command, you can delete any \*PRDLOD object whether or not you have authority to the \*PRDLOD object.

Top

---

## Parameters

Keyword	Description	Choices	Notes
PRDLOD	Product load, or	<i>Qualified object name</i>	Optional, Positional 1
	Qualifier 1: Product load, or	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *CURLIB</i>	
PRDID	Product ID	<i>Character value</i>	Optional
OPTION	Product option	1-99, *BASE	Optional
LODID	Load ID	<i>Character value, *CODEDFT</i>	Optional
RLS	Release level	<i>Character value, <u>*ONLY</u></i>	Optional

Top

---

## Product load, or (PRDLOD)

Specifies the name of the product load being deleted.

The possible library values are:

### \*CURLIB

The current library for the job is used to locate the product load.

### *library-name*

Specify the name of the library where the product load is located.

The possible product load value is:

### *product-load-name*

Specify the name of the product load being deleted.

**Note:** You must specify either the product ID, option, load ID, and release, or the product load name and the library.

Top

---

## Product (PRDID)

Specifies the 7-character identifier (ID) of the product for which a product load is being deleted.

Top

---

## Option (OPTION)

Specifies the product option for which a product load is being deleted.

The possible values are:

### **\*BASE**

The product load is part of the base option of the product.

### *product-option-number*

Specify the option number of the product load to be deleted. Valid values range from 1 through 99.

Top

---

## Load ID (LODID)

Specifies the ID of the product load to be deleted.

The possible values are:

### **\*CODEDFT**

The default code load ID, 5001, is used.

### *product-load-ID*

Specify the product load ID of the product load to be deleted.

Top

---

## Release level (RLS)

Specifies the version, release, and modification level of the product load to be deleted.

The possible values are:

### **\*ONLY**

The release level is determined by searching the system for a product load specified on the PRDID, OPTION, and LODID parameters. This value can be specified only when one release of the product is installed on the system.

### *version-release-modification*

Specify the version, release, and modification level of the product load being deleted.

Top

---

## Examples

```
DLTPRDL0D PRDL0D(MYLIB/MYLOD)
```

This command deletes the product load MYLOD in library MYLIB.

---

## Error messages

### \*ESCAPE Messages

#### CPF358A

Release not valid.





---

## Delete Remote PTF (DLTRMPTF)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Delete Remote Program Temporary Fix (DLTRMPTF) command allows a service provider to remotely delete PTFs from a service requester's system.

### Note:

A change request is automatically submitted that can be viewed to determine the status of this command. You can use the Work with Submitted Change Request (WRKSBMCRQ) command to monitor the status.

### Restrictions:

1. The delete function of this command is only supported when the service requester has Managed System Services licensed program installed.
2. If NODL value is specified, the node list must contain only entries that have an address type equal to \*SNA.

Top

---

## Parameters

Keyword	Description	Choices	Notes
PTFID	PTF description	Values (up to 50 repetitions): <i>Element list</i>	Required, Positional 1
	Element 1: PTF identifier	<i>Character value, *ALL</i>	
	Element 2: Product	<i>Character value, *ONLY</i>	
	Element 3: Release level	<i>Character value</i>	
DESTSRVRQS	Destination service requester	Single values: *ALL, * <u>SELECT</u> , *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Communications name</i>	
NODL	Node list name	<i>Qualified object name</i>	Optional
	Qualifier 1: Node list name	<i>Name, *NONE</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Top

---

## PTF description (PTFID)

Specifies the PTF that is to be deleted. A maximum of 50 PTF identifiers can be specified.

The possible PTF identifier values are:

### *ptf-identifier*

Specify a 7-character PTF identifier.

**\*ALL** Deletes all PTFs for the specified product.

The possible product identifier values are:

**\*ONLY**

Specifies that the PTF identifier is associated with only one product.

***product-id***

Specify the product of the PTF. The product must be specified when PTF identifiers are not unique across products or the PTF identifier is \*ALL.

The possible release level of product value is:

***release(VxRxMx)***

Specify the release level of the specified product. The format is VxRxMx.

Top

---

## Destination service requester (DESTSRVRQS)

Specifies the service requester where this PTF function is to be performed. A single service requester or a list of service requesters can be specified.

The possible network identifier values are:

**\*SELECT**

Shows a list of service requesters. From the list, you can select one or more service requesters. \*SELECT is not valid when the command is used in a batch environment.

**\*ALL** Specifies that this PTF function should occur on all of the service requesters that are defined for this service provider.

**\*NONE**

The service requesters are identified in a node list object.

**\*NETATR**

The remote network identifier of the service requester is the same as the network identifier of this system.

***remote-network-identifier***

Specify the remote network identifier of the service requester.

The possible control point values are:

This element is not valid when the network identifier is \*ALL, \*SELECT, or \*NONE.

***remote-control-point***

Specify the remote control point name of the service requester.

Top

---

## Node list name (NODL)

Specifies the node list object name that contains a list of service requesters where this PTF function is to be performed.

The possible values are:

**\*NONE**

The service requesters are identified in the DESTSRVRQS parameter.

The possible values are:

\***LIBL** Searches all of the libraries in the user and in the system portions of the job's library list for the node list object.

\***CURLIB**  
Searches for the node list object in the library for the current job.

*library-name*  
Specify the name of the library to be searched.

The possible node list name value is:

*node-list-name*  
Specify the node list name which contains the list of service requesters where this PTF function is to be performed.

Top

---

## Examples

### Example 1: Deleting a PTF

```
DLTRMTPTF PTFID((1234567 *ONLY V5R2M0))
```

This command shows how to delete a PTF on one or more service requesters. A list will be shown from which you select one or more service requesters.

### Example 2: Deleting a PTF on All Service Requesters

```
DLTRMTPTF PTFID((1234567 *ONLY V5R2M0)) DESTSRVRQS(*ALL)
```

This command shows how to delete a PTF from all service requesters supported by this service provider.

Top

---

## Error messages

None

Top



---

## Delete Submitted CRQ (DLTSBMCRQ)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Delete Submitted Change Request (DLTSBMCRQ) command allows you to delete one or more submitted change requests.

**Restriction:** You can only delete a change request if you are the submitter, or have \*JOBCTL special authority. Only change requests with a status of **Ended** can be deleted.

Top

---

### Parameters

Keyword	Description	Choices	Notes
CRQ	Change request	<i>Element list</i>	Required, Positional 1
	Element 1: Name	<i>Generic name, name, *ALL</i>	
	Element 2: Sequence number	1-999999, <u>*ALL</u>	
DAYS	Days	0-999, <u>30</u>	Optional

Top

---

### Change request (CRQ)

Specifies the change request name and sequence number to be deleted.

The possible change request name values are:

**\*ALL** Deletes all qualified change requests.

*change-request-name*

Specify the name of the change request to be deleted.

*generic\*-change-request-name*

Specify a generic change request name. A generic name is a character string containing one or more characters followed by an asterisk (\*). The change requests matching the string or beginning with the string is deleted.

The possible sequence number values are:

\*ALL Selects all qualified change requests.

*sequence-number*

Specify the sequence number of the change request to be deleted.

Top

---

## Days (DAYS)

Specifies the number of days old a change request must be before it is deleted. Change requests that have been ended for the specified number of days or more are deleted. The DAYS parameter is ignored for requests to delete a specific change request.

**30** Deletes change requests 30 days old or older.

*number-of-days-old*

Specify the number of days a change request has had a status of ENDED.

Top

---

## Examples

### Example 1: Deleting All Ended Change Requests

```
DLTSBMCRQ CRQ(*ALL) DAYS(0)
```

This command shows how to delete all ended change requests.

### Example 2: Deleting a Submitted Change Request

```
DLTSBMCRQ CRQ(CHG4444 123)
```

This command shows how to delete the submitted change request CHG4444 with a sequence number of 123.

### Example 3: Deleting All Change Requests

```
DLTSBMCRQ CRQ(CHG* 1) DAYS(15)
```

This command shows how to delete all change requests with names starting with CHG and having a sequence number of 1 that are 15 days old or older.

Top

---

## Error messages

None

Top

## Delete SMG Object (DLTSMGOBJ)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Delete System Manager Object (DLTSMGOBJ) command provides the capability of deleting an OS/400 object on a managed system.

### Note:

A change request is automatically submitted which can be viewed to determine the status of this command. A message is returned identifying the name of the change request.

### Restrictions:

1. Only OS/400 program objects or file members such as \*CL, \*REXX, and \*UNSPEC can be run.
2. The \*LIBL object library cannot be specified.
3. The object cannot reside in the QTEMP library.
4. If a node list (NODL) value is specified, the node list can only contain entries that have a value of \*SNA for the address type.

Top

## Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Object	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *CURLIB</i>	
OBJTYPE	Object type	<i>Character value</i>	Required, Positional 2
MBR	Member	<i>Name, *ALL, *FIRST, *LAST</i>	Optional
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: *NONE Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CPNAME	Managed system node names	Single values: *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Communications name</i>	

Top

---

## Object (OBJ)

Specifies the qualified name that is to be deleted.

The possible library values are one of the following:

### **\*CURLIB**

The current library for the job is used to locate the object.

### *library-name*

Specify only the library named in this parameter is searched.

The possible object name value is:

### *object-name*

Specify only characters A through Z, 0 through 9, dollar sign, at sign, and number sign are used in the object name.

Top

---

## Object type (OBJTYPE)

Specifies the object.

### *object-type*

Specify the object type.

Top

---

## Member (MBR)

Specifies the OS/400 file member name. It is ignored unless the object type is \*FILE.

**\*ALL** The entire file is deleted. \*ALL must be used if the file is not a physical file.

### **\*FIRST**

The first member is deleted. The member name is determined when the activity is run.

### **\*LAST**

The last member is deleted. The member name is determined when the activity is run by the date the member is added to the physical file.

### *member-name*

Specify the name of the member to be deleted.

Top

---

## Managed systems node list (NODL)

Specifies the node list name that contains a list of systems which are the destinations for the request. This parameter cannot be specified if the control point name (CPNAME parameter) is also specified.

The possible value is:

### **\*NONE**

The systems on which this activity is to be performed are not identified by a node list. Individual control point names must be specified.

The possible values are:



**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

**\*CURLIB**

The current library for the job is used to locate the node list object.

*library-name*

Specify that only the library named in the parameter is searched.

The possible node list name value is:

*node-list-name*

Specify the node list object name containing the list of systems on which the command is performed.

Top

---

## Managed system node names (CPNAME)

Specifies the managed systems APPN control point names on which this request is to be performed. Control point names cannot be specified if the node list (NODL) parameter is specified.

The possible value is:

**\*NONE**

The systems on which this request is to be performed are not identified individually. A node list must be specified.

The possible values are:

**\*NETATR**

The local system network ID is used. This is useful when the node being specified is in the same network as the local system.

*network-identifier*

Specify the managed system APPN network identifier on which the request is performed.

The possible control point name value is:

*control-point-name*

Specify the managed system APPN control point name on which the request is performed.

Top

---

## Examples

### Example 1: Delete Object from Systems in Node List

```
DLTSMGOBJ OBJ(PAYROLL/BACKUP) OBJTYPE(*FILE)
          NODL(MYLIB/ALLSYS)
```

This command deletes the payroll backup file on all systems listed in node list ALLSYS.

### Example 2: Delete Object from Specific Systems

```
DLTSMGOBJ OBJ(PGMLIB/PROCPGM) OBJTYPE(*PGM)
          CPNAME((*NETATR SYS1) (*NETATR SYS2))
```

This command deletes the program PROCPGM from the library PGMLIB on systems SYS1 and SYS2 in the local network.

[Top](#)

---

## Error messages

None

[Top](#)

## Display Submitted CRQs (DSPSBMCRQ)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Display Submitted Change Requests (DSPSBMCRQ) command allows you to display or print a list of submitted change requests.

### Restriction:

Only the change requests submitted by you display. If you have \*JOBCTL special authority, all selected change requests display.

Top

## Parameters

Keyword	Description	Choices	Notes
CRQ	Change request	<i>Element list</i>	Optional, Positional 1
	Element 1: Name	<i>Generic name, name, *ALL</i>	
	Element 2: Sequence number	1-999999, *ALL	
STATUS	Status	*ALL, *SBM, *ACTIVE, *ENDED	Optional
HIGHENDCDE	Highest end code	0-99, *ANY, *SUCCESS, *FAIL, *NOTRUN	Optional
LASTENDCDE	Last end code	0-99, *ANY, *SUCCESS, *FAIL, *NOTRUN	Optional
CRQD	Change request description	<i>Qualified object name</i>	Optional
	Qualifier 1: Change request description	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *ALL</i>	
SBM	Submitter	<i>Name, *ALL</i>	Optional
PERIOD	Period	<i>Element list</i>	Optional
	Element 1: Start time and date	<i>Element list</i>	
	Element 1: Start time	<i>Time, *AVAIL</i>	
	Element 2: Start date	<i>Date, *BEGIN, *CURRENT</i>	
	Element 2: End time and date	<i>Element list</i>	
	Element 1: Ending time	<i>Time, *AVAIL</i>	
	Element 2: Ending date	<i>Date, *END, *CURRENT</i>	
PRBID	Problem identifier	<i>Character value, *ALL</i>	Optional
PRBORG	Problem origin	<i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *ALL, *NETATR</i>	
	Element 2: Control point	<i>Communications name, *ALL, *NETATR</i>	
OUTPUT	Output	<i>_, *PRINT</i>	Optional

Top

---

## Change request (CRQ)

Specifies the qualified change request name.

The possible change request name values are:

**\*ALL** Display all the qualified change requests.

*change-request-name*

Specify that only the specified change requests display.

*generic\*-change-request-name*

Specify a generic change request name. A generic name is a character string containing one or more characters followed by an asterisk (\*). Only the change request names matching or beginning with the string displays.

The possible sequence number values are:

**\*ALL** Display all selected change requests regardless of sequence number.

*sequence-number*

Specify that only the change requests with a sequence number display.

Top

---

## Status (STATUS)

Specifies the status of the change request to display.

**\*ALL** Display all qualified change requests regardless of status.

**\*SBM** Display only the change requests with a status of Submitting.

**\*ACTIVE**

Display only the change requests with a status of Active.

**\*ENDED**

Display the change requests with a status of Ended.

Top

---

## Highest end code (HIGHENDCDE)

Specifies the highest end code or range for which change requests are displayed.

**\*ANY** Display all change requests regardless of the highest end code.

**\*FAIL** Display all change requests with the highest end code between 10 and 89.

**\*SUCCESS**

Display all change requests with the highest end code between 0 and 9.

**\*NOTRUN**

Display all change requests with the highest end code between 90 and 99.

*highest-end-code*

Specify the highest end code for which to display change requests. The valid values are 0 through 99.

Top

---

## Last end code (LASTENDCDE)

Specifies that the last change request end code or range for which change requests are displayed.

**\*ANY** Displays all change requests regardless of the last end code.

**\*FAIL** Display all change requests with the last end code between 10 and 89.

**\*SUCCESS**  
Display all change requests with the last end code between 0 and 9.

**\*NOTRUN**  
Display all change requests with the last end code between 90 and 99.

### *last-end-code*

Specify the last end code for which to display change requests. Valid values range from 0 through 99.

Top

---

## Change request description (CRQD)

Specifies the qualified name of the change request description object.

The possible values are:

**\*ALL** Search all libraries in the system for the specified change request description.

### *library-name*

Specify that only the library specified in this parameter be searched.

The possible values for change request description names are:

**\*ALL** Display all change requests regardless of their change request descriptions.

### *change-request-description*

Specify that only change requests based on the specified change request description be displayed.

### *generic\*-change-request-description*

Specify a generic change request description name. A generic name is a character string containing one or more characters followed by an asterisk(\*). Display only change requests based on the change request description names matching or beginning with the string.

Top

---

## Submitter (SBM)

Specifies the user profile that submitted the change request.

**\*ALL** Specify that all selected change requests submitted by the user profile name specified are displayed.

### *user-profile-name*

Specify that only the change requests submitted by the user profile name specified display.

Top

---

## Period (PERIOD)

Specifies a period of time for which the change requests are displayed. The following values can be coded in this parameter, which contains two list of two elements each:

The possible starting time values are:

One of following is used to specify the starting time at which or after which the change request must have been submitted. Change requests submitted before the specified time and date are not displayed.

### \*AVAIL

The change requests that are available for the specified starting date are displayed.

### *start-time*

Specify the starting time for the specified starting date that indicates the change requests to be displayed.

The possible starting date values are:

One of the following is used to specify the starting date on which or after which the change requests must have been submitted. Entries submitted before the specified date are not displayed.

### \*BEGIN

Change requests are displayed beginning with the date that the first change request was submitted. If \*BEGIN is specified, then any time value other than \*AVAIL for start-time is ignored.

### \*CURRENT

The change requests submitted today between the specified starting and ending times are displayed.

### *start-date*

Specify the starting date for which change requests are displayed.

The possible ending time values are:

One of the following is used to specify the ending time before which the change requests must be submitted.

### \*AVAIL

The change requests that are available for the specified ending date are displayed.

### *end-time*

Specify the ending time for the specified ending date that determines the change requests to be displayed.

The possible ending date values are:

One of the following is used to specify the ending date before which or on which the change requests must have been submitted.

### \*END

The last day on which change requests have been submitted is the last day for which the change requests are displayed. If \*END is specified, then any value other than \*AVAIL for ending time is ignored.

### \*CURRENT

The change requests submitted today between the specified starting and ending times are displayed.

### *end-date*

Specify the ending date for which change requests are displayed.

---

## Problem identifier (PRBID)

Specifies the problem identifier associated with the change requests to be displayed.

**\*ALL** Specifies that change requests are displayed regardless of the associated problem ID.

*problem-identifier*

Specify the associated problem ID for which the change request is displayed.

---

## Problem origin (PRBORG)

Specifies the originating system of the problem ID.

The possible network ID values are:

**\*ALL** Display change requests regardless of the problem origin network ID.

**\*NETATR**

Display change requests with a problem origin network ID that is the same as the network ID in the system network attributes.

*network-ID*

Specify a problem origin network ID for which change requests are displayed.

The possible control point name values are:

**\*ALL** Display change requests regardless of the problem origin control point name.

**\*NETATR**

Display change requests with a control point name that is the same as the control point name defined in the system network attributes.

*control-point-name*

Specify a problem origin control point name for which change requests are displayed.

---

## Output (OUTPUT)

Specifies whether the output from the command shows at the requesting work station, or prints with the job's spooled output.

**\*** Output requested by an interactive job is shown on the display. Batch job output prints with the job's spooled output.

**\*PRINT**

Prints the output with the job's spooled output.

---

## Examples

### Example 1: Displaying All Change Requests by Generic Name

```
DSPSBMCRQ (CRQ* *ALL)
```

This command displays all change requests with names that start with CRQ.

### Example 2: Displaying All Change Requests That Failed

```
DSPSBMCRQ STATUS(*ENDED) LASTENDCDE(*FAIL)
```

This command displays all change requests that ended in a failed state.

### Example 3: Displaying Change Requests by a Specific User

```
DSPSBMCRQ SBM(JOE)  
PERIOD((*AVAIL 08/31/02) (*AVAIL *CURRENT))
```

This command displays all change requests submitted by the user profile JOE beginning 31 August 2002 to the current date.

### Example 4: Displaying Change Requests for a Specific Change Request Description

```
DSPSBMCRQ CRQD(CRQD001)
```

This command displays all submitted change requests based on the CRQD001 change request description.

[Top](#)

---

## Error messages

None

[Top](#)



---

## Display Submitted CRQ Act (DSPSBMCRQA)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Display Submitted Change Request Activities (DSPSBMCRQA) command displays the list of activities for the selected change request.

### Restriction:

Only messages for a change request that you submitted are displayed. If you have \*JOBCTL special authority, you can view the messages for any change requests.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CRQ	Change request	<i>Element list</i>	Required, Positional 1
	Element 1: Name	<i>Name</i>	
	Element 2: Sequence number	1-999999	
STATUS	Status	*ALL, *HELD, *NOTSTARTED, *SCHEDULED, *WAIT, *READY, *STARTED, *SENT, *NOTSENT, *RUNNING, *ENDING, *ENDED	Optional
ENDCODE	End code	0-99, *ANY, *FAIL, *SUCCESS, *NOTRUN	Optional
OUTPUT	Output	*, *PRINT	Optional

Top

---

## Change request (CRQ)

Specifies the change request name and the sequence number of the activity list to display.

The possible change request name value is:

*change-request-name*

Specify the name of the of the change request for which activities are displayed.

The possible sequence number value is:

*sequence-number*

Specify the change request sequence number.

Top

---

## Status (STATUS)

Specifies the status criteria for displaying activities.

Possible values are:

**\*ALL** All change request activities display regardless of status.

**\*HELD**  
Displays the activities with a status of Held.

**\*NOTSTARTED**  
Displays the activities with a status of Not Started.

**\*SCHEDULED**  
Displays the activities with a status of Scheduled.

**\*WAIT**  
Displays the activities with a status of Wait.

**\*READY**  
Displays the activities with a status of Ready.

**\*STARTED**  
Displays only the activities with a status of Started.

**\*RUNNING**  
Displays only the activities with a status of Running.

**\*ENDING**  
Displays only the activities with a status of Ending.

**\*ENDED**  
Displays only the activities with a status of Ended.

**\*SENT**  
Displays only the activities with a status of Sent.

**\*NOTSENT**  
Displays only the activities with a status of Notsent.

Top

---

## End code (ENDCODE)

Specifies the end code for the change request activities displayed.

Possible values are:

**\*ANY** Display activities regardless of end codes.

**\*SUCCESS**  
Displays all activities with the highest end code between 0 and 9.

**\*FAIL** Displays all activities with the highest end code between 10 and 89.

**\*NOTRUN**  
Displays all activities with the highest end code between 90 and 99.

### *end-code*

Specify the end code to use. Displays activities with the specified end code or an end code greater than the one specified. The valid values are 0 through 99.

Top

---

## Output (OUTPUT)

Specifies whether the command output files displays at the requested work station, or prints with the spooled output file.

\*  
\_ Output requested by an interactive job is shown on the display. Batch job output prints with the job's spooled output.

**\*PRINT**

The output prints with the job's spooled output.

[Top](#)

---

## Examples

DSPSBMCRQA CRQ(CHG001 123456)

Displays the activities of the change request named CHG001 with the sequence number 123456.

[Top](#)

---

## Error messages

None

[Top](#)



## Display Submitted CRQ Messages (DSPSBMCRQM)

Where allowed to run: All environments (\*ALL)  
 Threadsafes: No

Parameters  
 Examples  
 Error messages

The Display Submitted Change Request Messages (DSPSBMCRQM) command displays or prints a list of messages for one or more submitted change requests.

### Restriction:

You can only display messages for change requests you submitted, unless you have \*JOBCTL authority. In this case, you can display messages for any change request.

Top

## Parameters

Keyword	Description	Choices	Notes
CRQ	Change request	<i>Element list</i>	Optional, Positional 1
	Element 1: Name	<i>Generic name, name, *ALL</i>	
	Element 2: Sequence number	<i>1-999999, *ALL</i>	
ACTIVITY	Activity name	<i>Name, *ALL, *LAST</i>	Optional
CPNAME	Managed system node names	<i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *ALL, *NETATR</i>	
	Element 2: Control point	<i>Character value, *ALL, *NETATR</i>	
SEV	Severity code	<i>0-99, 0</i>	Optional
PERIOD	Period	<i>Element list</i>	Optional
	Element 1: Start time and date	<i>Element list</i>	
	Element 1: Start time	<i>Time, *AVAIL</i>	
	Element 2: Start date	<i>Date, *BEGIN, *CURRENT</i>	
	Element 2: End time and date	<i>Element list</i>	
	Element 1: Ending time	<i>Time, *AVAIL</i>	
	Element 2: Ending date	<i>Date, *END, *CURRENT</i>	
OUTPUT	Output	<i>*, *PRINT</i>	Optional

Top

## Change request (CRQ)

Specifies the qualified change request name used to display messages.

The possible change request name values are:

**\*ALL** Displays the messages for all change requests matching the selection criteria of the other parameters.

***change-request-name***

Specify a change request name for which the messages are displayed.

***generic\*-change-request-name***

Specify the generic change request name for which messages are displayed. A generic name is a string containing one or more characters followed by an asterisk (\*).

The possible sequence number values are:

**\*ALL** Displays messages for the change request specified regardless of the sequence number.

***sequence-number***

Specify a change request sequence number that identifies the change request for which messages are displayed.

Top

---

## Activity name (ACTIVITY)

Specifies the name of the activity for which the messages are displayed.

The possible values are:

**\*ALL** Display all change request messages regardless of the activity name.

**\*LAST**

Display the message for the last activity that runs. The actual stored name is \*LAST.

***activity-name***

Specify a change request activity for which you want to display messages.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if a node list is specified.

The possible network identifier values are:

**\*ALL** Messages are displayed regardless of the destination network ID.

**\*NETATR**

Only change request activity messages for which the destination network ID is the same as the one defined in the network attributes for this system, are displayed.

***network-identifier***

Specify a network ID. Only change request messages with the destination network ID and with the control point name specified, are displayed.

The possible control point values are:

**\*ALL** Messages display regardless of the destination node control point name.

**\*NETATR**

Only change request activity messages for which the destination control point name is the same as the one defined in the system network attributes, are displayed.

### *control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed. For NetView Distribution Management Agents, the control point name is the change control client which supports numeric characters (0-9) in the first position of control point names that are valid in other platforms.

Top

---

## Severity code (SEV)

Specifies the severity level for which messages are displayed.

0 All messages in the message queue specified are shown.

### *severity-code*

Specify a value, 0 through 99. Displays messages with a severity code equal to or greater than the value specified.

Top

---

## Period (PERIOD)

Specifies a period of time for which the change requests or messages are displayed. The following values can be coded in this parameter, which contains two list of two elements each:

The possible starting time values are:

One of following is used to specify the starting time at which or after which the change request message must have been logged. Change requests logged before the specified time and date are not displayed.

### \*AVAIL

The change requests messages that are available for the specified starting date are displayed.

### *start-time*

Specify the starting time for the specified starting date that indicates the change requests to be displayed.

The possible starting date values are:

One of the following is used to specify the starting date on which or after which the change requests messages must have been logged. Messages logged before the specified date are not displayed.

### \*BEGIN

The change request messages from the beginning of the log are displayed. If \*BEGIN is specified, then any time value other than \*AVAIL for start-time is ignored.

### \*CURRENT

The change request messages logged today between the specified starting and ending times are displayed.

### *start-date*

Specify the starting date for which change request messages are displayed.

The possible ending time values are:

One of the following is used to specify the ending time before which the change request messages must have been logged.

### **\*AVAIL**

The change request messages that are available for the specified ending date are displayed.

### ***end-time***

Specify the ending time for the specified ending date that determines the change request messages to be displayed.

The possible ending date values are:

One of the following is used to specify the ending date before which or on which the change request messages must have been logged.

**\*END** The last day on which change request messages have been logged is the last day for which the change request messages are displayed. If \*END is specified, then any value other than \*AVAIL for ending time is ignored.

### **\*CURRENT**

The change request messages submitted today between the specified starting and ending times are displayed.

### ***end-date***

Specify the ending date for which change request messages are displayed.

Top

---

## **Output (OUTPUT)**

Specifies that the output from the command is shown at the work station requested, or prints with the job's spooled output.

**\*** An interactive job's output shows on the display. Batch job output prints with the job's spooled output.

### **\*PRINT**

The output prints with the job's spooled output.

Top

---

## **Examples**

### **Example 1: Displaying All Messages for a Change Request**

```
DSPSBMCRQM CRQ(CHG0001 000001)
```

This example shows how to display all messages for change request CHG0001 with sequence number 000001.

### **Example 2: Displaying All Severity 30 Messages for a Change Request**

```
DSPSBMCRQM CRQ(CHG0001 000001) ACTIVITY(APLYPTF) SEV(30)
```

This example shows how to display all change request messages for CRQ00001 000001 activity APLYPTF with severity greater than, or equal to 30.

### **Example 3: Displaying All Messages for a Specified Period**

```
DSPSBMCRQM CRQ(CHGRQS*) CPNAME(*NETATR CHICAGO)  
PERIOD((*AVAIL 08/31/02) (*AVAIL *CURRENT ))
```



This example displays all messages for change requests starting with CHGRQS and that have a destination of CHICAGO. Only the messages logged between 8/31/02 and the current date are displayed.

[Top](#)

---

## Error messages

None

[Top](#)



---

## Display Service Provider Attr (DSPSRVPVDA)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Display Service Provider Attributes (DSPSRVPVDA) command displays the configuration attributes of the service provider.

The Change Service Provider Attributes (CHGSRVPVDA) command can be used to change the service provider attributes.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

DSPSRVPVDA

This command displays the configuration attributes of the service provider.

[Top](#)

---

### Error messages

None

[Top](#)



---

## End Submitted CRQ Activity (ENDSBMCRQA)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The End Submitted Change Request Activity (ENDSBMCRQA) command allows you to end one or more change request activities.

### Restriction:

You must be either the submitter of the change request or have \*JOBCTL special authority.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CRQ	Change request	<i>Element list</i>	Required, Positional 1
	Element 1: Name	<i>Name</i>	
	Element 2: Sequence number	1-999999	
ACTIVITY	Activity name	<i>Name, *ALL, *LAST</i>	Optional
CPNAME	Managed system node name	<i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *ALL, *NETATR</i>	
	Element 2: Control point	<i>Character value, *ALL, *NETATR</i>	
OPTION	How to end	<i>*CNTRLD, *IMMED, *FRCFAIL</i>	Optional

Top

---

## Change request (CRQ)

Specifies the change request name and the change request sequence number for the activities that are to be ended.

The possible change request name values are:

### *change-request-name*

Specify the change request name of the activities to be ended.

The possible sequence number values are:

### *sequence-number*

Specify the change request sequence number of the activities to be ended.

Top

---

## Activity name (ACTIVITY)

Specifies the activity that is ended.

**\*ALL** End all activities of the specified change request.

**\*LAST**

End the activity named \*LAST. This is not the last activity added to the change request description object. This is the last activity to be run after the change request is submitted.

*activity-name*

Specify the name of the activity to end.

Top

---

## Managed system node name (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if the node list (NODL) parameter is specified.

The possible network identifier values are:

**\*ALL** All activities for the specified control point name are ended.

**\*NETATR**

Only network ID activities that match the ones defined in the network attributes for this system are ended.

*network-identifier*

Specify a network ID. Only activities for the network ID and the control point name specified are ended.

The possible control point values are:

**\*ALL** All activities for the specified network ID are ended.

**\*NETATR**

Only activities for the control point name that match the one defined in the network attributes for this system are ended.

*control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed. For NetView Distribution Management Agents, the control point name is the change control client which supports numeric characters (0-9) in the first position of control point names that are valid in other platforms.

Top

---

## How to end (OPTION)

Specifies how the activity is ended.

The possible values are:

**\*CNTRLD**

The activity is ended in a controlled manner. Activities that have not started to run are ended. Activities that have already started (Started or Running) only end if they do not leave partial results. The end code returned is 30.

**\*IMMED**

The activity is to be ended as soon as possible even if partial results occur. Activities that have not yet started to run are ended. Activities that have already started (Started or Running) can have their process interrupted which can cause partial results. The end code returned is 35.

### **\*FRCFAIL**

The activity is ended immediately. If the activity cannot be ended, it is marked as ended with an end code of 39. Other activities conditioned on this activity are evaluated to determine if they are ready to run.

[Top](#)

---

## **Examples**

```
ENDSBMCRQA  CRQ(CHG001 456)  ACTIVITY(*ALL)  CPNAME(*ALL)
```

This command ends all of the activities for the change request named CHG001 with sequence number 456 for all the nodes in a controlled manner.

[Top](#)

---

## **Error messages**

None

[Top](#)





---

## End System Manager (ENDSYSMGR)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The End System Manager (ENDSYSMGR) command ends the job in the QSYSWRK subsystem that run the System Manager functions.

### Restrictions:

1. You must have \*JOBCTL authority to use the end command.
2. Public authority for this command is \*EXCLUDE.

Top

---

## Parameters

Keyword	Description	Choices	Notes
OPTION	How to end	* <u>CNTRLD</u> , *IMMED	Optional, Positional 1
DELAY	Delay time, if *CNTRLD	0-999999, * <u>NOLIMIT</u>	Optional, Positional 2

Top

---

## How to end (OPTION)

Specifies that the change request manager monitor job and the electronic customer support control job have a controlled ending, or are to be ended immediately.

### \*CNTRLD

Specify that the change request manager monitor job and the electronic customer support control job ends in a controlled manner.

### \*IMMED

Specify that the change request manager monitor job and the electronic customer support control job ends immediately.

Top

---

## Delay time, if \*CNTRLD (DELAY)

Specifies the delay time before ending is immediate.

### \*NOLIMIT

The change request manager monitor job and the electronic customer support control job continue processing until the current activity processing is complete. Both jobs name a controlled ending with the maximum value (999999 seconds) specified.

### *delay-time*

Specify that the change request manager monitor job and the electronic customer support control job end immediately after the delay time.

---

## Examples

### Example 1: Ending Jobs

```
ENDSYSMGR
```

This command ends the jobs in the QSYSWRK subsystem that run the System Manager function in a controlled manner.

### Example 2: Ending Jobs Immediately

```
ENDSYSMGR OPTION(*IMMED)
```

This command ends the jobs in the QSYSWRK subsystem that run the System Manager function in an immediate manner.

---

## Error messages

None

---

## Generate License Key (GENLICKEY)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Generate License Key (GENLICKEY) command generates a license key to enable users to access a product or a feature of a product. This key is specific to the product and system information entered in this command. The resulting key will be a combination of 18 characters and numbers, A-F and 0-9. To run this command, the product definition of the product you are generating the key for must exist on the system.

This command also adds the license information to the license repository. The keys are saved in the repository to keep a history of all the keys generated. You can display the repository to see what keys were generated for a specific product or system. To work with the license repository, you can use the DSPLICKEY, ADDLICKEY, and RMVLICKEY commands.

You can use the Add License Key Information (QLZAADDK) API or the Add Product License Information (QLZADDLI) API to add license information. For more specific details about these APIs, see the API topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

**Restriction:** This command is shipped with public \*EXCLUDE authority.

Top

---

### Parameters

Keyword	Description	Choices	Notes
PRDID	Product ID	<i>Character value</i>	Required, Positional 1
LICTRM	License term	<i>Character value</i>	Required, Positional 2
FEATURE	Feature	<i>Character value</i>	Required, Positional 3
VNDPWD	Vendor password	<i>Simple name, *NONE</i>	Required, Positional 4
SERIAL	System serial number	<i>Character value</i>	Required, Positional 5
PRCGRP	Processor Group	<i>Character value, *ANY</i>	Optional, Positional 6
EXPDATE	Expiration date	<i>Date, *NONE</i>	Optional
USGLMT	Usage limit	<i>0-999999, 1, *NOMAX</i>	Optional
VNDDTA	Vendor data	<i>Character value, *NONE</i>	Optional
OUTPUT	Output	<i>*, *PRINT, *LICKEYFILE</i>	Optional
LICKEYFILE	License key file	<i>Qualified object name</i>	Optional
	Qualifier 1: License key file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Keyword	Description	Choices	Notes
LICKEYMBR	License key member	<i>Element list</i>	Optional
	Element 1: Member	<i>Name, *FIRST</i>	
	Element 2: Replace or add records	<i>*ADD, *REPLACE</i>	

Top

---

## Product ID (PRDID)

Specifies the 7-character identifier (ID) of the product for which the license key is to be generated.

Top

---

## License term (LICTRM)

Specifies the license term for which license key is to be generated.

The possible values are:

### *license-term*

Specify the license term of the product in one of the following formats:

- V $x$
- V $x$ R $y$
- V $x$ R $y$ M $z$

where values for  $x$  and  $y$  range from 0 through 9, and values for  $z$  range from 0 through 9 and from A through Z. A license term of V $x$  means the authorized usage limit is valid for the entire version of the product or feature. A license term of V $x$ R $y$  means the authorized usage limit is valid for the entire release of the product or feature. A license term of V $x$ R $y$ M $z$  means the authorized usage limit is valid only for a modification of the product.

Top

---

## Feature (FEATURE)

Specifies the feature of the product specified on the PRDID parameter for which the license key is to be generated.

The possible values are:

### *feature*

Specify the feature for which a license key is to be generated. Valid values range from 5001 through 9999.

Top

---

## Vendor password (VNDPWD)

Specifies the software vendor's password. This password will be encrypted and stored with the product. It is used to validate the generate license key request and must be the same password you use on the Add Product License Information (ADDPRDLICI) command.

The possible values are:

**\*NONE**

There is no vendor password needed to generate a license key.

*vendor-password*

Specify the vendor password. It must begin with an alphabetic character (A through Z, \$, #, or @) followed by no more than 9 alphanumeric characters (A through Z, 0 through 9, \$, #, @, or \_).

Top

---

## System serial number (SERIAL)

Specify the system serial number of the system the license key is being generated for.

The possible values are:

*system-serial-number*

Specifies the system serial number of the system the license key is being generated for.

Top

---

## Processor Group (PRCGRP)

Specifies the processor group of the system the license key is being generated for.

The possible values are:

**\*ANY** The license key being generated will work with any processor group.

*processor-group*

Specifies the processor group of the system the license key is being generated for. Use the Work with License Information (WRKLICINF) command on the system the key is being generated for to display the processor group.

Top

---

## Expiration date (EXPDATE)

Specifies the date the product license expires. After this date, users over the default usage limit will not be allowed to access the product or feature. A new license key must be obtained from the software vendor to allow further use of the product.

The possible values are:

**\*NONE**

There is no expiration date for the product or feature.

*expiration-date*

Specify the date that the product or feature expires.

Top

---

## Usage limit (USGLMT)

Specifies the maximum number of users (usage limit) for the product or feature for which the license key is being generated. For concurrent usage, specifies the maximum number of jobs allowed to access the product or feature at the same time. For registered usage, this is the maximum number of license users that can be registered to use this product or feature.

The possible values are:

1 One user is allowed to access the product or feature.

**\*NOMAX**

The number of users is not limited.

*usage-limit*

Specifies the maximum number of users for this product or feature. Valid values range from 0 through 999999.

Top

---

## Vendor data (VNDDTA)

Specifies an 8-character data field to be defined and used by the vendor.

The possible values are:

\*NONE

There is no vendor data for this product or feature.

*vendor-data*

Specifies 8-characters that help define further information on the product or feature.

Top

---

## Output (OUTPUT)

Specifies whether the license key that is generated is displayed in a completion message, printed in a spooled file, or saved in a file.

The possible values are:

\* The license key is only shown in a completion message.

**\*PRINT**

The license key is printed in a spooled file.

**\*LICKEYFILE**

The license key information is saved to the file specified on the LICKEYFILE parameter. This file can then be used as a key file for Distributed Systems License Option (DSLO) tapes.

Top

---

## License key file (LICKYFILE)

Specifies the file in which the license key information is saved. If this file does not exist, the system creates it. If it does exist, it must be in the format of QSYS/QALZKEY.

The possible values are:

**\*LIBL** The library list is used to locate the file in which to save the license key.

**\*CURLIB**

The current job library is used to locate the file in which to save the license key. If no library is specified as the current library, the QGPL library is used.

*library-name*

Specify the name of the library where the license key is to be saved.

The possible file name value is:

*file-name*

Specify the name of the file into which the license key information is to be saved.

Top

---

## License key member (LICKYMBR)

Specifies which member the license key information is saved into and how the information is saved.

**Note:**

You can also use the Generate License Key (QLZAGENK) API to create the license key information. See the API topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter> for more information about this API.

The possible member to receive output values are:

**\*FIRST**

The first member in the file receives the output. If LICKYMBR(\*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified on the LICKYFILE parameter.

*member-name*

Specify the name of the file member that receives the output. If the name of the file member is specified and the member does not exist, the system creates it. If the member already exists, you have the option of either adding a new record to the end of the existing member or clearing the member and then adding the new records.

The option to perform on member values are:

**\*ADD** If the member exists, the system adds the new record to the end of the existing records.

**\*REPLACE**

If the member exists, the system clears it and then adds the new record.

Top

---

## Examples

```
GENLICKEY PRDID(2MYPROD) LICTRM(V5R2M0) FEATURE(5001)
          VNDPWD(PRODUCTPWD) EXPDATE(*NONE)
          SERIAL(1234567) PRCGRP(P30) USGLMT(25)
```

This command generates a license key for product 2MYPROD, license term V5R2M0 and feature 5001. The key allows 25 users to be authorized to use this product and there is no expiration date. The key allows the product to only be used on a system with a serial number of 1234567 and a processor group of 01 and a processor group of P30 or less.

[Top](#)

---

## Error messages

None

[Top](#)



---

## Hold Program Temporary Fix (HLDPTF)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Hold Program Temporary Fix (PTF) (HLDPTF) command holds a PTF save file (SAVF). When a PTF SAVF is put on hold, it is not automatically sent to a service requester by way of electronic customer support.

Top

---

### Parameters

Keyword	Description	Choices	Notes
PTF	PTF number	<i>Character value</i>	Required, Positional 1
LICPGM	Product	<i>Character value, *ALL</i>	Optional
RLS	Release level	<i>Character value, *ONLY</i>	Optional

Top

---

### PTF number (PTF)

Specifies which PTF to hold.

Top

---

### Product (LICPGM)

Specifies the product identifier (ID) of the PTF being held.

The possible values are:

**\*ALL** All products are searched for the PTF being held.

*product-ID*

Specify the product ID of the PTF being held.

Top

---

### Release level (RLS)

Specifies the release level of the PTF to be held.

The possible values are:

**\*ONLY**

The PTF exists for only one release level of the product.

*version-release-modification*

Specify the release level of the PTF being held.

---

## Examples

HLDPTF PTF(SI12345)

This command holds the save files of the PTF named SI12345. Holding SI12345 prevents it from being automatically distributed to a service requester.

---

## Error messages

### \*ESCAPE Messages

SMU1430

Duplicate PTF found.

---

## Hold Submitted CRQ Activity (HLDSBMCRQA)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Hold Submitted Change Request Activity (HLDSBMCRQA) command holds one or more change request activities.

### Restrictions:

1. Only activities that have a status of Wait, Scheduled, or Ready can be held.
2. You must be either the submitter of the change request, or have \*JOBCTL special authority.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CRQ	Change request	<i>Element list</i>	Required, Positional 1
	Element 1: Name	<i>Name</i>	
	Element 2: Sequence number	1-999999	
ACTIVITY	Activity name	<i>Name, *ALL, *LAST</i>	Optional
CPNAME	Managed system node name	<i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *ALL, *NETATR</i>	
	Element 2: Control point	<i>Character value, *ALL, *NETATR</i>	

Top

---

## Change request (CRQ)

Specifies the change request name and the change request sequence number of the activities held.

The possible change request name value is:

The possible values are:

*change-request-name*

Specify the change request name of the activities that are held.

The possible sequence number value is:

*sequence-number*

Specify the change request sequence number of the activities to be held.

Top

---

## Activity name (ACTIVITY)

Specifies the activity to hold.

**\*ALL** Hold all of the specified change request activities.

**\*LAST**

Hold the activity named \*LAST. This is not the last activity added to the change request description object. This is the last activity run after the change request is submitted.

*activity-name*

Specify the name of the activity to hold.

Top

---

## Managed system node name (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if the node list (NODL) parameter is specified.

The possible network identifier values are:

**\*ALL** All activities for the specified control point name are held.

**\*NETATR**

Specify that only activities for the network ID matching the one defined in the network attributes for this system are held.

*network-identifier*

Specify a network ID. Only activities for the network ID and the control point name specified are held.

The control point values are:

**\*ALL** All activities for the specified network ID are held.

**\*NETATR**

Only activities for the control point name that match the one defined in the network attributes for this system are held.

*control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed. For NetView Distribution Management Agents, the control point name is the change control client which supports numeric characters (0-9) in the first position of control point names that are valid in other platforms.

Top

---

## Examples

```
HLDSBMCRQA CRQ(CHG001 456)
```

This command holds all of the activities for the change request named CHG001 with sequence number 456 for all the nodes.

Top

---

## Error messages

None

Top

# Install Remote Product (INSRMPRD)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Install Remote Product (INSRMPRD) command provides the capability to install a product, packaged using the System Manager/400 licensed program, from the central site system, to one or more managed systems.

## Note:

A change request is automatically submitted which can be viewed to determine the status of this command. A message is returned identifying the name of the change request.

## Restrictions:

1. This command is shipped with public \*EXCLUDE authority.
2. To run this command, you must have the authority necessary to use the Add Product Change Request Activity (ADDPDCRQA) command.
3. If a NODL value is specified, the node list can only contain entries that have a value of \*SNA for the address type.
4. The Managed System Services licensed product and the OS/400 licensed product cannot be specified.

Top

## Parameters

Keyword	Description	Choices	Notes
PRDID	Product ID	Character value	Required, Positional 1
RLS	Release level	Character value, *ONLY	Optional
OPTION	Product option	1-99, *BASE	Optional
LODTYPE	Product load type	*ALL, *CODE, *LNG	Optional
LODID	Load ID	2900-2999, *ALL, *CODE	Optional
TGTRLS	Target release	*CURRENT, *PRV, V5R1M0, V5R2M0, V5R3M0	Optional
NODL	Managed systems node list	Element list	Optional
	Element 1: Node list	Single values: *NONE Other values: Qualified object name	
	Qualifier 1: Node list	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
CPNAME	Managed system node names	Single values: *NONE Other values (up to 50 repetitions): Element list	Optional
	Element 1: Network identifier	Communications name, *NETATR	
	Element 2: Control point	Communications name	
KEEPCLGE	Keep catalog entry	*NO, *YES	Optional

Top

---

## Product ID (PRDID)

Specifies the 7-character identifier of the product to be installed.

### *product-ID*

Specify the 7-character product ID that is installed.

Top

---

## Release level (RLS)

Specifies which version, release, and modification level of the product is used.

The possible values are:

### \*ONLY

The release level is determined by searching the system for a product definition for the given product ID. The release level is taken from the product definition. This value is not valid if more than one product definition exists for the same product ID.

### *version-release-modification*

Specify the release level in the format VxRxMy, where Vx is the version number, and My is the modification number. Valid values for *x* are the numbers 0 through 9. Valid values for *y* are the numbers 0 through 9 and the letters A through Z.

Top

---

## Product option (OPTION)

Specifies which of the optional parts of the product given in the PRDID parameter are used.

The possible values are:

### \*BASE

Only the base part of the product is used.

### *product-option-number*

Specify the option number for the product load being saved. Valid values range from 1 through 99.

Top

---

## Product load type (LODTYPE)

Specifies the product load objects being used.

The possible values are:

\*ALL Code and language objects specified on the LODID parameter are used.

### \*CODE

The object associated with this product load are used.

\*LNG The objects associated with the national language version (NLV) identified on the LODID parameter are used.

Top

---

## Load ID (LODID)

Specifies the load identifier used.

The possible values are:

**\*ALL** All languages for this product option are saved.

**\*CODE**

The code load is used.

### *product-load-ID*

Specify when LOCTYPE(\*LNG) or LOCTYPE(\*ALL) is specified, the load ID must be one of the valid IBM national language versions and be specified in the form 29xx, where xx is any value from zero through 9.

Top

---

## Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object. This parameter is ignored for objects with global names that are in the SystemView distribution repository or for actions other than send or retrieve.

The possible values are:

**\*CURRENT**

The object is used on the release of the operating system currently running on your system. If V5R3M0 is running on your system, \*CURRENT means that you intend to use the object on a system with V5R3M0 installed. The object can also be used on a system with any later release of the operating system installed.

**\*PRV** The object is intended for a system that is at the previous release level compared to the local system.

**Note:** Modification levels are not supported. To specify the previous release with a modification level other than 0, such as V4R1M4, specify the value for the release rather than the special value \*PRV.

### *release-level*

Specify the release level in the VxRxMx format. The object is used on a system with the specified release or with any later release of the operating system installed.

Valid values depend on the current version, release, and modification level and they change with each new release.

Top

---

## Managed systems node list (NODL)

Specifies that the node list parameter is the object name that contains a list of systems which are the destinations for the activity. It cannot be specified if control point name (CPNAME) parameter is also specified.

The possible values are:

**\*NONE**

The systems on which this activity is to be performed are not identified by a node list. Individual control point names must be specified.

The possible library values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

**\*CURLIB**

The current library for the job is used to locate the node list object.

***library-name***

Specify the name of the library to be searched.

The possible node list name value is:

***node-list-name***

Specify the node list object name containing the list of systems on which the activity is to be performed.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if a node list is specified. If ACTION(\*RTV) is used, only one control point name can be specified.

**\*NONE**

The systems on which this activity is performed are not identified individually. A node list must be specified.

The possible network identifier values are:

**\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

***network-identifier***

Specify the APPN network identifier of the managed system on which the activity is to be performed.

The possible control point name value is:

***control-point-name***

Specify the APPN control point name of the managed system on which the activity is to be performed.

Top

---

## Keep catalog entry (KEEPCLGE)

Specifies if the distribution catalog entry and its associated save file corresponding to the product is kept on the specified system.

The possible values are:

**\*NO** The catalog entry and associated save file are not kept.

**\*YES** The catalog entry and associated save file are kept.

Top



---

## Examples

### Example 1: Installing a Product

```
INSRMTPRD  PRDID(1ACCOUN)  RLS(V5R2M0)  OPTION(*BASE)
           LODTYPE(*ALL)  LODID(*ALL)  CPNAME((*NETATR SYS1))
```

This command installs 1ACCOUN product with the release V5R2M0 of the base option for both the program and the language parts on the iSeries SYS1.

### Example 2: Installing an Option

```
INSRMTPRD  PRDID(1CHECKS)  RLS(V5R2M0)  OPTION(10)
           LODTYPE(*ALL)  LODID(*ALL)  CPNAME((*NETATR SYS2))
```

This command installs option 10 of the 1CHECKS product, release V5R2M0, for both the program and the language parts on the iSeries SYS2.

[Top](#)

---

## Error messages

None

[Top](#)



---

## Order Supported Product PTFs (ORDSPTPTF)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Order Supported Product PTFs (ORDSPTPTF) command orders program temporary fixes (PTFs) for supported products. This is done by comparing the PTFs on the system with the PTFs available from IBM service support, and then ordering the PTFs that are:

1. Not loaded
2. Not applied
3. Not on order
4. Not in a PTF save file (SAVF) in the QGPL library

The PTF order request created by this command is sent to IBM service support.

This command detects any PTFs on the system that IBM now considers defective. This command also detects PTFs not on the system that IBM considers HIPER (High-Impact Pervasive). In either case, a message indicating PTF number and recommended actions is sent to the service provider message queue.

Top

---

### Parameters

Keyword	Description	Choices	Notes
DELIVERY	Delivery method	*ANY, *LINKONLY	Required, Positional 1
LICPGM	Product	Character value, <u>*ALL</u>	Optional, Positional 2

Top

---

### Delivery method (DELIVERY)

Specifies how PTFs are delivered.

This is a required parameter.

The possible values are:

**\*ANY** Any method of delivery is acceptable. If the order that was sent to IBM service support is small enough, the PTF order will arrive through electronic customer support. If the PTF order is larger than the acceptable electronic customer support limit, PTFs arrive by tape.

**\*LINKONLY**

Only delivery through electronic customer support is acceptable. If the PTF order is larger than the acceptable electronic customer support limit, a message is sent indicating that the PTF delivery failed.

Top

---

## Product (LICPGM)

Specifies which licensed programs are having PTFs ordered for them.

The possible values are:

**\*ALL** PTFs are ordered for all licensed programs.

*licensed-program-ID*

Specify the licensed program that is having PTFs ordered for it.

Top

---

## Examples

### Example 1: Ordering PTFs for All Supported Products

```
ORDSPTPTF DELIVERY(*ANY)
```

This command orders PTFs for all supported products. If the order is small enough, the PTFs arrive by way of electronic customer support; otherwise, they arrive by tape.

### Example 2: Ordering PTFs for a Licensed Program

```
ORDSPTPTF DELIVERY(*LINKONLY) LICPGM(5722SS1)
```

This command orders PTFs for the licensed program named 5722SS1 (OS/400). The PTFs arrive by way of electronic customer support.

Top

---

## Error messages

### \*ESCAPE Messages

**CPF8C07**

A parameter is not valid.

Top

---

## Package Product Option (PKGPRDOPT)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Package Product Option (PKGPRDOPT) command packages one or more product loads for a specified product option enabling the loads to be saved using the Save License Program (SAVLICPGM) command.

The following notes provide information on how the command works.

### Notes:

1. To package a product load that has folders, you must be enrolled in the system distribution directory and you must have \*ALL authority to the folders.  
If you have authority for this command, you can package any product created using the System Manager program regardless of whether you have authority to the objects for the product.
2. The only object types that the PKGPRDOPT command package is on is the Save Object list Application Program Interface (API). Other object types are ignored.

Top

---

## Parameters

Keyword	Description	Choices	Notes
PRDID	Product ID	<i>Character value</i>	Required, Positional 1
OPTION	Product option	1-99, *BASE	Required, Positional 2
ALWAPICHG	Allow change by program	*SAME, *NO	Required, Positional 3
LODID	Load ID	<i>Character value</i> , *ALL, *CODEDFT	Optional
RLS	Release level	<i>Character value</i> , *ONLY	Optional
REPACKAGE	Package again	*NO, *YES	Optional

Top

---

## Product ID (PRDID)

Specifies the product identifier (ID) of the product option being packaged. The ID must be seven characters in length.

Top

---

## Product option (OPTION)

Specifies the product option being packaged.

The possible values are:

**\*BASE**

The base option is packaged.

***product-option-number***

Specify the option number of the product to be packaged. Valid values range from 1 through 99.

Top

---

## **Allow change by program (ALWAPICHG)**

Specifies whether to prevent changes to the description of each object processed by this command.

The possible values are:

**\*SAME**

The 'allow change by program' attribute of the objects being packaged is not changed by this command.

**\*NO**

The object description for each object packaged cannot be changed by the QLICOBJD application program interface (API) or the Change Product Object Description (CHGPRDOBJD) command.

Top

---

## **Load ID (LODID)**

Specifies the load ID of the product loads being packaged.

The possible values are:

**\*ALL**

All product loads for which a product load object exists are packaged.

If only a code load exists, the code load is packaged.

**\*CODEDFT**

The default code load, 5001, is packaged.

***product-load-ID***

Specify a language load ID or the default code load, 5001, for a code load.

Top

---

## **Release level (RLS)**

Specifies the version, release, and modification level of the product option.

The possible values are:

**\*ONLY**

The release level is determined by searching the system for a product definition for the given product ID. The release level is taken from the product definition. This value is not valid if more than one product definition exists for the same product ID.

***version-release-modification***

Specify the version, release, and modification level of the product being packaged.

Top

---

## Package again (REPACKAGE)

Specifies whether to allow the command to package a product option which has already been packaged.

The possible values are:

**\*NO** Product options which have been packaged are not repackaged.

**\*YES** Package all specified product options regardless of their current state.

Top

---

## Examples

```
PKGPRDOPT PRDID(9XYZ123) OPTION(*BASE) ALWAPICHG(*SAME)
```

This command packages the base option of the product 9XYZ123.

Top

---

## Error messages

### \*ESCAPE Messages

#### **CPF0CB2**

Product identifier &1 not valid.

#### **CPF0CEB**

Product loads not packaged.

#### **CPF0CE2**

&9 product loads packaged, &10 product loads not packaged.

#### **CPF0CE3**

&9 product loads packaged, &10 product loads not packaged. See job log.

#### **CPF0CE7**

Product &1 release &2 not packaged.

#### **CPF0CFC**

Product definition not found.

#### **CPF0CFD**

Code load not found for product &1 release &2 option &3.

#### **CPF0CFF**

Multiple releases available.

#### **CPF0C4B**

Product availability object &2/&1 recovery required.

#### **CPF0C4C**

Cannot allocate object &1 in library &2.

#### **CPF0C4D**

Error occurred while processing object &1 in library &2.

#### **CPF8122**

&8 damage on library &4.

#### **CPF8191**

Product definition &4 in &9 damaged.

**CPF8193**

Product load object &4 in &9 damaged.

**CPF9012**

Start of document interchange session not successful for &1.

**CPF9032**

Document interchange session not started.

**CPF9803**

Cannot allocate object &2 in library &3.

**CPF9830**

Cannot assign library &1.

[Top](#)



---

## Release Program Temporary Fix (RLSPTF)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Release Program Temporary Fix (RLSPTF) command releases a Program Temporary Fix (PTF) save file (SAVF) that was previously on hold. When a PTF SAVF is released, it can be automatically sent to a service requester by way of electronic customer support.

Top

---

### Parameters

Keyword	Description	Choices	Notes
PTF	PTF number	<i>Character value</i>	Required, Positional 1
LICPGM	Product	<i>Character value, *ALL</i>	Optional
RLS	Release level	<i>Character value, *ONLY</i>	Optional

Top

---

### PTF number (PTF)

Specifies which PTF is to be released.

Top

---

### Product (LICPGM)

Specifies the product identifier (ID) of the PTF being released.

The possible values are:

**\*ALL** All products are searched for the PTF being released.

*product-ID*

Specify the product ID of the PTF being released.

Top

---

### Release level (RLS)

Specifies the release level of the PTF to be released.

The possible values are:

**\*ONLY**

The PTF exists for only one release level of the product.

*version-release-modification*

Specify the release level of the PTF to be released.

---

## Examples

RLSPTF PTF(SI12345)

This command releases the save files of the PTF named SI12345. Releasing SI12345 allows it to be automatically distributed to a service requester.

---

## Error messages

### \*ESCAPE Messages

SMU1430

Duplicate PTF found.

---

## Release Submitted CRQ Activity (RLSSBMCRQA)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Release Submitted Change Request Activity (RLSSBMCRQA) command releases one or more change request activities.

### Restrictions:

1. Only activities in Held or Not Started status can be released.
2. You must be either the submitter of the change request or have \*JOBCTL special authority.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CRQ	Change request	<i>Element list</i>	Required, Positional 1
	Element 1: Name	<i>Name</i>	
	Element 2: Sequence number	1-999999	
ACTIVITY	Activity name	<i>Name, *ALL, *LAST</i>	Optional
CPNAME	Managed system node name	<i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *ALL, *NETATR</i>	
	Element 2: Control point	<i>Character value, *ALL, *NETATR</i>	

Top

---

## Change request (CRQ)

Specifies the change request name and the change request sequence number whose activities are to be released.

The possible change request name values are:

### *change-request-name*

Specify the name of the change request whose activities are to be released.

The possible sequence number values are:

### *sequence-number*

Specify the change request sequence number of the activities that are to be released.

Top

---

## Activity name (ACTIVITY)

Specifies the activity to be released.

The possible values are:

**\*ALL** Release all activities of the specified change request.

**\*LAST**

Release the activity named \*LAST. This is not the last activity added to the change request description object. This is the last activity that runs after the change request is submitted.

*activity-name*

Specify the name of the activity to be released.

Top

---

## Managed system node name (CPNAME)

Specifies the destination node for an activity that should be released. The node consists of two elements which are the network identifier and the control point name.

The possible network identifier values are:

**\*ALL** All activities for the specified control point name are released.

**\*NETATR**

Only activities having network IDs that match the one defined in the network attributes for this system are released.

*network-identifier*

Specify a network ID. Only the activities having the network ID and control point name specified, are released.

The possible control point values are:

**\*ALL** All activities for the specified network ID are released.

**\*NETATR**

Only activities for the control point name matching the one defined in the network attributes for this system are released.

*control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed. For NetView Distribution Management Agents, the control point name is the change control client which supports numeric characters (0-9) in the first position of control point names that are valid in other platforms.

Top

---

## Examples

```
RLSSBMCRQA CRQ(CHG001 456) ACTIVITY(*ALL) CPNAME(*ALL)
```

This command releases all of the activities for the change request named CHG001 with sequence number 456 for all the nodes.

Top

---

## Error messages

None

[Top](#)



---

## Remove CRQD Activity (RMVCRQDA)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Remove Change Request Description Activity (RMVCRQDA) command removes an activity from a change request description.

**Restriction:** You must have \*CHANGE authority to the change request description object to remove an activity from it.

Top

---

### Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACTIVITY	Activity name	<i>Name, *LAST</i>	Required, Positional 2

Top

---

### Change request description (CRQD)

Specifies the name and library of the change request description where the activity is removed.

The possible library values are:

**\*LIBL** Locates the change request description in the library list. The first matching change request description is used.

**\*CURLIB**

Locates the change request description in the current library. If no library is specified as the job's current library, the QGPL library is used.

***library-name***

Specify that only the library specified is searched.

The possible change request description name values are:

***change-request-description-name***

Specify the names of the change requests description from which an activity is removed.

Top

---

## Activity name (ACTIVITY)

Specifies the activity to remove.

The possible values are:

### **\*LAST**

The activity named \*LAST is removed.

### *activity-name*

Specify the name of the activity to be removed.

Top

---

## Examples

```
RMVCRQDA CRQD(*LIBL/CHG001) ACTIVITY(ACT01)
```

This command removes the ACT01 activity from the first change request description found in the library list of the current interactive job with the name CHG001.

Top

---

## Error messages

None

Top



# Remove Remote PTF (RMVRMTPTF)

Where allowed to run: All environments (\*ALL)  
 Threadsafes: No

Parameters  
 Examples  
 Error messages

The Remove Remote Program Temporary Fix (RMVRMTPTF) command allows a service provider to remotely remove PTFs from a service requester's system. When using the RMVRMTPTF command, you can request an IPL of the service requester's system. The remove PTF and IPL can be scheduled to occur at a later date and time.

## Note:

A change request is automatically submitted that can be viewed to determine the status of this command. You can use the Work with Submitted Change Requests (WRKSCMCRQ) command to monitor the results.

## Restrictions:

1. The remove and IPL functions of this command are only supported when the service requester has Managed System Services licensed program installed.
2. If a NODL value is specified, the node list can only contain entries that have a value of \*SNA for the address type.
3. PTFs are always removed temporarily except VLIC PTFs which are removed permanently.
4. Dependent PTFs can be removed remotely. PTFs are removed in the order that they are listed on the command.

Top

## Parameters

Keyword	Description	Choices	Notes
PTFID	PTF description	Values (up to 300 repetitions): <i>Element list</i>	Required, Positional 1
	Element 1: PTF identifier	<i>Character value, *ALL</i>	
	Element 2: Product	<i>Character value, *ONLY</i>	
	Element 3: Release level	<i>Character value</i>	
DESTSRVRQS	Destination service requester	Single values: *ALL, *SELECT, *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Communications name</i>	
DLYRMV	Delayed PTF remove	*NO, *YES	Optional
NODL	Node list name	<i>Qualified object name</i>	Optional
	Qualifier 1: Node list name	<i>Name, *NONE</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
IPL	IPL remote system	*NO, *YES	Optional
PWRDWNOPT	Power down option	*CNTRL, *IMMED	Optional
PWRDWNDLY	Power down delay	1-65535, <u>3600</u>	Optional

Keyword	Description	Choices	Notes
RMTRMVTIME	Remote remove time	<i>Element list</i>	Optional
	Element 1: Time zone	<i>*LCLSYS, *MGDSYS</i>	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time, *CURRENT</i>	
	Element 2: Date	<i>Date, *CURRENT, *NEXT</i>	
RMTIPLTIME	Remote IPL time	<i>Element list</i>	Optional
	Element 1: Time zone	<i>*LCLSYS, *MGDSYS</i>	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time, *CURRENT</i>	
	Element 2: Date	<i>Date, *CURRENT, *NEXT</i>	

Top

---

## PTF description (PTFID)

Specifies the PTF that is to be removed. A maximum of 300 PTF identifiers can be specified.

The possible PTF identifier values are:

### *PTF-identifier*

A 7-character PTF identifier.

**\*ALL** Removes all PTFs for the specified product.

The product identifier values are:

### **\*ONLY**

Specifies that the PTF identifier is associated with only one product.

### *product-id*

Specify the product of the PTF. The product must be specified when PTF identifiers are not unique across products or the PTF identifier is **\*ALL**.

The release level of product values are:

### *release (VxRxMx)*

Specify the release level of the product. The format is VxRxMx.

Top

---

## Destination service requester (DESTSRVRQS)

Specifies the service requester where this PTF function is to be performed. A single service requester or a list of service requesters can be specified.

The possible network identifier values are:

### **\*SELECT**

Shows a list of service requesters. From the list, you can select one or more service requesters. **\*SELECT** is not valid when the command is used in a batch environment.

**\*ALL** Specifies that this PTF function should occur on all of the service requesters that are defined for this service provider.

**\*NONE**

The service requesters are identified in a node list object.

**\*NETATR**

The remote network identifier of the service requester is the same as the network identifier of this system.

***remote-network-identifier***

Specify the remote network identifier of the service requester.

The possible control point value is:

This element is not valid when the network identifier is \*ALL, \*SELECT, or \*NODE.

***remote-control-point***

Specify the remote control point name of the service requester.

Top

---

## Delayed PTF remove (DLYRMV)

Specifies how PTFs are removed. Remove immediate PTFs at the time the activity runs on the specified system, or later during the next IPL.

The possible values are:

**\*NO** Removes an immediate PTF at the time the activity runs. If the PTF is marked delayed, it is not removed until the next unattended IPL.

**\*YES** Removes both immediate or delayed PTFs during the next unattended IPL.

Top

Specifies the node list object name that contains a list of service requesters where this PTF function is to be performed.

The possible values are:

**\*NONE**

The service requesters are identified in the DESTSRVRQS parameter.

The possible library values are the following:

**\*LIBL** Searches all of the libraries in the user and system portions of the job's library list for the node list object.

**\*CURLIB**

Searches for the node list object in the library for the current job.

***library-name***

Specify the name of the library to be searched.

The possible value for the node list name is:

***node-list-name***

Specify the node list name which contains the list of service requesters where this PTF function is to be performed.

Top

---

## IPL remote system (IPL)

Specifies if an IPL of the service requester's system should occur.

The possible values are:

**\*NO** An IPL of the service requester system should not occur.

**\*YES** An IPL of the service requester system should occur after all the PTFs listed on the command are successfully removed.

Top

---

## Power down option (PWRDWNOPT)

Specifies how to manage the end of active jobs during the IPL processing at the service requester system. This parameter is valid only if IPL(\*YES) is specified.

### **\*CNTRLD**

All jobs end when the time specified in Power Down Delay (PWRDWDLY) parameter expires. Any current jobs that are running perform end of end of batch processing.

### **\*IMMED**

Jobs end immediately.

Top

---

## Power down delay (PWRDWNDLY)

Specifies the amount of time in seconds that the system allows a controlled end to be performed. If the end of job routines are not finished in the specified power down delay, any remaining jobs are ended immediately. This parameter is only valid if IPL(\*YES) and if PWRDWNOPT(\*CNTRLD) are specified.

**3600** The amount of time in which to complete a controlled end of processing is limited to 3600 seconds.

### **delay-time**

Specify the maximum amount of delay time, in seconds, in which a controlled end can be performed. Valid values range from 1 through 65535 (65,535 seconds).

Top

---

## Remote remove time (RMTRMVTIME)

Specifies the date and time when this PTF function can occur on the service requester. The current date and time values and next date are determined, when this PTF function is processed on this system, based on this system's date and time.

The possible time zone values are:

### **\*LCLSYS**

Specifies the remote start time in the time zone of this system.

### **\*MGDSYS**

Specifies the remote start time in the time zone of the service requester's system.

The possible start after time values are:

### **\*CURRENT**

Starts this PTF function on the service requester system at any time on or after this PTF function request is processed on this system.

#### ***start-after-time***

Specify the time when this PTF function can be started on the service requester. The time can be entered as 4 or 6 digits, such as hhmm or hhmmss where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

### **\*CURRENT**

Starts this PTF function on the service requester system on any date on or after the date this PTF function request is processed on this system.

### **\*NEXT**

Starts this PTF function on the service requester system on any date after the date this PTF function is processed on this system.

#### ***start-after-date***

Specify the date when this PTF function can start on the service requester system. The date must be specified in the job date format.

#### **Note:**

The special values \*CURRENT and \*NEXT cannot be specified for the date and time if the time zone value \*MGDSYS is specified.

Top

---

## **Remote IPL time (RMTIPLTIME)**

Specifies the date and time when the IPL can occur on the service requester. The current date and time values and next date are determined, when this IPL request is processed on this system, based on this system's date and time.

The possible time zone values are:

### **\*LCLSYS**

Specifies the remote start time in the time zone of this system.

### **\*MGDSYS**

Specifies the remote start time in the time zone of the service requester's system.

The possible start after time values are:

### **\*CURRENT**

Starts the IPL on the service requester system at any time on or after the IPL request is processed on this system.

#### ***start-after-time***

Specify the time when the IPL can be started on the service requester. The time can be entered as 4 or 6 digits, such as hhmm or hhmmss where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

### **\*CURRENT**

Starts the IPL on the service requester system on any date on or after the date this IPL request is processed on this system.

### **\*NEXT**

Starts the IPL on the service requester system on any date after the date this IPL request is processed on this system.

### ***start-after-date***

Specify the date when the IPL can start on the service requester system. The date must be specified in the job date format.

**Note:** The special values \*CURRENT and \*NEXT cannot be specified for the date and time if the time zone value \*MGDSYS is specified.

Top

---

## **Examples**

### **Example 1: Removing a PTF**

```
RMVRMTPTF PTFID((123456 *ONLY V5R2M0)) DESTSRVRQS(*SELECT)
```

Remove a PTF from one or more selected service requesters.

### **Example 2: Removing a PTF on a Certain Date**

```
RMVRMTPTF PTFID((*ALL PRODUCT V5R2M0))
DESTSRVRQS((*NETATR CHICAG01) (*NETATR CHICAG02))
RMTAPYTIME((*MGDSYS ('10:00:00' '10/31/02'))
IPL(*YES)
RMTIPLTIME((*MGDSYS ('01:00:00' '11/01/02'))
```

This command removes the PTFs on the service requesters starting at 10:00 a.m. on October 31, 2002. If all the PTFs were removed successfully, the service requester's system IPL starts at 1:00 a.m. on November 1, 2002.

### **Example 3: Doing an IPL After Removing a PTF**

```
RMVRMTPTF PTFID((987654 *ONLY V5R2M0))
DESTSRVRQS(*SELECT ') IPL(*YES)
RMTIPLTIME(*LCLSYS (*CURRENT 12/24/2002))
PWRDWNPT(*IMMED)
```

This command removes the PTF on the selected service requesters. If the PTF is removed successfully, the IPL on the remote system starts at the current time on December 24, 2002. The times specified are in the time zone of the local system.

### **Example 4: Removing a PTF with Two Corequisite PTFs and One Prerequisite PTF**

```
RMVRMTPTF PTFID((SF00001 *ONLY V5R2M0)) DESTSRVRQS(*SELECT)
RMTRMVTIME(*LCLSYS (*CURRENT *CURRENT)) IPL(*NO)
```

This command removes PTF number SF00001. This PTF has 2 corequisite PTFs and one prerequisite PTF. PTFs SF00002 and SF00003 are corequisite PTFs for PTF SF00001. PTF SF00004 is a prerequisite for PTF SF00001. The system will remove PTFs SF0001, SF00002, and SF00003. However, the prerequisite PTF will not be removed. The system will determine all corequisite PTFs if they were not specified. No IPL will be performed, and this action applies the PTFs on the service requesters. The times specified are in the time zone of the local system.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF358A

PTF &1-&2 cannot be applied or removed.

[Top](#)





---

## Restore APAR Data (RSTAPARDTA)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Restore APAR Data (RSTAPARDTA) command updates a problem log entry with the specified Authorized Problem Analysis Report (APAR) library name. This is the problem log entry saved with the APAR data on the service requester and is associated with the APAR library. If the problem log entry does not exist, one is created on the service provider. All spooled files saved using the Save APAR Data (SAVAPARDTA) command are re-spooled on the service provider when RSTAPARDTA is run.

Top

---

### Parameters

Keyword	Description	Choices	Notes
FRMAPARLIB	From APAR Library	<i>Name</i>	Required, Positional 1

Top

---

### From APAR Library (FRMAPARLIB)

Specifies the library where to restore the APAR data. This library must contain the data from a specific APAR.

Top

---

### Examples

```
RSTAPARDTA FRMAPARLIB(MYAPARLIB)
```

This command restores on the service provider the APAR data in library MYAPARLIB.

Top

---

### Error messages

#### \*ESCAPE Messages

##### CPF2182

Not authorized to library &1.

##### CPF39FA

Problem &1 &2 &3 not found

##### CPF39FB

Library &1 does not exist

##### CPF39FD

No APAR data restored from library &1

**CPF39F2**

Cannot allocate library &1

**CPF39F5**

Query of problem &1 failed

**CPF39F9**

Problem &1 &2 &3 in use

[Top](#)

---

## Retrieve Product (RTVPRD)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Retrieve Product (RTVPRD) command provides the capability to retrieve a product, packaged with the iSeries product packaging support, from a managed system.

### Note:

A change request is automatically submitted which can be viewed to determine the status of this command. A message is returned identifying the name of the change request.

### Restrictions:

1. This command is shipped with public \*EXCLUDE authority.
2. The Managed System Services licensed program and the OS/400 product cannot be specified.
3. A product can only be retrieved from a single managed system.

Top

---

## Parameters

Keyword	Description	Choices	Notes
PRDID	Product ID	<i>Character value</i>	Required, Positional 1
RLS	Release level	<i>Character value, *ONLY</i>	Optional
OPTION	Product option	1-99, *BASE	Optional
LODTYPE	Product load type	*ALL, *CODE, *LNG	Optional
LODID	Load ID	2900-2999, *ALL, *CODE	Optional
TGTRLS	Target release	*CURRENT, *PRV, V5R1M0, V5R2M0, V5R3M0	Optional
CPNAME	Managed system node names	<i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Communications name</i>	

Top

---

## Product ID (PRDID)

Specifies the 7-character identifier of the product to be retrieved.

### *product-ID*

Specify the 7-character product ID that is used in the activity.

Top

---

## Release level (RLS)

Specifies which version, release, and modification level of the product is used.

The possible values are:

**\*ONLY**

Only one version, release, and modification level is used for the product option.

***release-level***

Specify the release level in the format VxRxMy, where Vx is the version number, and My is the modification number. Valid values for x are the numbers 0 through 9. Valid values for y are the numbers 0 through 9 and the letters A through Z.

Top

---

## Product option (OPTION)

Specifies which of the optional parts of the product given in the PRDID parameter are used.

The possible values are:

**\*BASE**

Only the base part of the product is used.

***product-option-number***

Specify the option number for the product load being saved. Valid values range from 1 through 99.

Top

---

## Product load type (LODTYPE)

Specifies the product load objects being used.

The possible values are:

**\*ALL** Code and language objects specified on the LODID parameter are used.

**\*CODE**

The program objects associated with this product load are used.

**\*LNG** Specify the objects associated with the NLV identified on the LODID parameter are used.

Top

---

## Load ID (LODID)

Specifies the load identifier used.

The possible values are:

**\*ALL** All languages for this product option are saved.

**\*CODE**

The code load is used.

### *product-load-ID*

Specify when LODTYPE(\*LNG) or LODTYPE(\*ALL) is used, the load ID must be one of the valid IBM national language versions and must be specified in the form 29xx. The value of x can be 0 through 9.

Top

---

## Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object. This parameter is ignored for objects with global names that are in the SystemView distribution repository or for actions other than send or retrieve.

### \*CURRENT

The object is used on the release of the operating system currently running on your system. If V5R3M0 is running on your system, \*CURRENT means that you intend to use the object on a system with V5R3M0 installed. The object can also be used on a system with any later release of the operating system installed.

**\*PRV** The object is intended for a system which is at the previous release level compared to the local system.

**Note:** Modification levels are not supported. To specify a previous release with a modification level other than 0, such as V4R1M4, specify the value for the release rather than the special value \*PRV.

### *release-level*

Specify the release level in the VxRxMx format. The object is used on a system with the specified release or with any later release of the operating system installed.

Valid values depend on the current version, release, and modification level, and they change with each new release.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if the node list (NODL) is specified.

The possible values are:

### \*NONE

Specify the systems on which this activity is to be performed. They are not identified individually. A node list must be specified.

### **\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

### *network-identifier*

Specify the APPN network identifier of the managed system on which the activity is to be performed.

The possible control point name value is:

### *control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed.

---

## Examples

### Example 1: Retrieving the Program and Language Parts

```
RTVPRD  PRDID(1ACCOUN)  RLS(V5R2M0)  OPTION(*BASE)
        LODTYPE(*ALL)  LODID(*ALL)  CPNAME((*NETATR SYS1))
```

This command retrieves the program and language parts for the base option of product 1ACCOUN with release V5R2M0 from iSeries server SYS1.

### Example 2: Retrieving an Option

```
RTVPRD  PRDID(1CHECKS)  RLS(V5R2M0)  OPTION(10)
        LODTYPE(*ALL)  LODID(*ALL)  CPNAME((*NETATR SYS2))
```

This command retrieves the option 10 of the 1CHECKS product, release V5R2M0, for both the program and the language parts from the iSeries server SYS2.

---

## Error messages

None

---

## Retrieve Program Temporary Fix (RTVPTF)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Retrieve Program Temporary Fix (RTVPTF) command allows a service provider to retrieve remote PTFs and cover letters from a service requester's system.

### Note:

A change request is automatically submitted that can be viewed to determine the status of this command. You can use the Work with Submitted Change Request (WRKSBMCRQ) command to monitor the status.

### Restriction:

The retrieve function of this command is only supported when the service requester has the Managed System Services licensed program installed.

Top

---

## Parameters

Keyword	Description	Choices	Notes
PTFID	PTF description	Values (up to 50 repetitions): <i>Element list</i>	Required, Positional 1
	Element 1: PTF identifier	<i>Character value</i>	
	Element 2: Product	<i>Character value, *ONLY</i>	
	Element 3: Release level	<i>Character value</i>	
DESTSRVRQS	From service requester	Single values: <b>*SELECT</b> Other values: <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Communications name</i>	
PTFPART	PTF parts	<b>*ALL</b> , *PTF, *CVRLTR	Optional
CVRLTRLNG	Cover letter language	Values (up to 40 repetitions): <i>Character value, *SRVPVDA</i>	Optional

Top

---

## PTF description (PTFID)

Specifies the PTF that is to be retrieved. A maximum of 50 PTF identifiers can be specified.

The possible PTF identifier values are:

*ptf-identifier*

Specify a 7-character PTF identifier.

The possible product identifier values are:

**\*ONLY**

The PTF identifier is associated with only one product.

### *product-id*

Specify the product of the PTF. The product must be specified when PTF identifiers are not unique across products.

The possible release level of product values are:

### *release(VxRxMx)*

Specify the release level of the product. The release level must be specified when PTF identifiers are not unique across products. The format is VxRxMx.

Top

---

## From service requester (DESTSRVRS)

Specifies the service requester where the PTF should be retrieved. Only one service requester can be specified or selected.

The possible network identifier values are:

### \*SELECT

Shows a list of service requesters. From the list, the user can select a service requester from where the PTF is retrieved. \*SELECT is not valid when the command is used in a batch environment.

### \*NETATR

The remove network identifier of the service requester is the same as the network identifier of this system.

### *remote-network-identifier*

Specify the remote network identifier of the service requester.

The possible control point value is:

### *remote-control-point*

Specify the remote control point name of the service requester.

Top

---

## PTF parts (PTFPART)

Specifies if PTFs or cover letters should be retrieved.

The possible values are:

\*ALL Retrieve the PTF and its cover letter in the language specified.

\*PTF Retrieve only the PTF.

### \*CVRLTR

Retrieve only the cover letter in the language specified.

Top

---

## Cover letter language (CVLRLNG)

Specifies the cover letter language that should be retrieved.

The possible values are:



## \*SRVPVDA

The value set in the Change Service Provider Attributes (CHGSRVPVDA) command is used.

### *cover-letter-language-ID*

Specify the cover letter language code. Prompting (F4) on this field provides a list of all possible cover letter language codes.

#### **Note:**

If the specified cover letter language is not found on the service requester's system, a default cover letter language of 2924 which is upper and lower case English, or 2950 which is upper case English, is retrieved if available.

Top

---

## **Examples**

### **Example 1: Retrieving a PTF and Cover Letter**

```
RTVPTF  PTFID(SF00001)  DESTSRVRQS(*NETATR CHICAG01)
```

This command retrieves a PTF and a cover letter from a service requester. The cover letter language to be retrieved is defined as a service provider attribute.

### **Example 2: Retrieving a Cover Letter**

```
RTVPTF  PTFID(123456)  DESTSRVRQS(*SELECT)
        PTFPART(*CVRLTR)  CVRLTRLNG(2962)
```

This command retrieves a cover letter from a selected service requester in the 2962 language.

Top

---

## **Error messages**

None

Top



---

## Retrieve SMG Object (RTVSMGOBJ)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Retrieve System Manager Object (RTVSMGOBJ) command provides the capability of retrieving an OS/400 object from a managed system.

### Note:

A change request is automatically submitted which can be viewed to determine the status of this command. A message is returned identifying the name of the change request.

### Restrictions:

1. The object cannot reside in the QTEMP library.
2. An object can only be retrieved from a single managed system.

Top

---

## Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Object	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *CURLIB</i>	
OBJTYPE	Object type	<i>Character value</i>	Required, Positional 2
MBR	Member	<i>Name, *ALL, *FIRST, *LAST</i>	Optional
CPNAME	Managed system node names	<i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Communications name</i>	
TGTRLS	Target release	<i>*CURRENT, *PRV, V5R1M0, V5R2M0, V5R3M0</i>	Optional
REPLACE	Replace object	<i>*NO, *YES</i>	Optional
DTACPR	Data compression	<i>*NONE, *SNA</i>	Optional

Top

---

## Object (OBJ)

Specifies the qualified name to be retrieved.

The possible library values are one of the following:

### \*CURLIB

The current library for the job is used to locate the object.

### *library-name*

Specify that only the library named in this parameter is to be searched.

The possible object name value is:

### *object-name*

Specify that only characters A through Z, 0 through 9, dollar sign, at sign, and number sign are used in the object name.

Top

---

## Object type (OBJTYPE)

Specifies the object to be used.

### *object-type*

Specify the object type.

### **\*FILEDATA**

A file member is sent without the file attributes.

Top

---

## Member (MBR)

Specifies the OS/400 physical file member name. It is ignored unless the object type is \*FILE or \*FILEDATA.

**\*ALL** The entire physical file is to be retrieved. \*ALL must be used if the file is not a physical file.

### **\*FIRST**

Only the first member is retrieved. The member name is determined when the activity is run.

### **\*LAST**

Only the last member is retrieved. The member name is determined when the activity is run by the date the member was added to the physical file.

### *member-name*

Specify names of the member to be retrieved.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed system from which this activity is to be retrieved.

### **\*NETATR**

The network ID of the local system is used. This is useful when the node specified is in the same network as the local system.

### *network-identifier*

Specify the managed system APPN network identifier which the activity is to be performed.

The possible value for the control point name is:

### *control-point-name*

Specify the managed system APPN control point name on which the activity is to be performed.

---

## Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object. This parameter is ignored for objects with global names that are in the SystemView distribution repository or for actions other than send or retrieve.

### \*CURRENT

The object is used on the release of the operating system currently running on your system. If V5R3M0 is running on your system, \*CURRENT means that you intend to use the object on a system with V5R3M0 installed. The object can also be used on a system with any later release of the operating system installed.

**\*PRV** The object is intended for a system which is at the previous release level compared to the local system.

### *release-level*

Specify the release level in the VxRxMx format. The object is used on a system with the specified release or with any later release of the operating system installed.

Valid values depend on the current version, release, and modification level, and they change with each new release.

Top

---

## Replace object (REPLACE)

Specifies if the object should be replaced if it already exists. This parameter cannot be specified for actions other than for \*SND, \*SNDRUN, and \*RTV.

**\*NO** An error is returned if the object already exists.

**\*YES** The object is replaced if it already exists.

Top

---

## Data compression (DTACPR)

Specifies that data is compressed when retrieving. Compression is valid only for object type \*FILEDATA. SNA compression with a blank prime compression character is performed.

**\*SNA** The file data is compressed when retrieved.

### **\*NONE**

The file data is not compressed when retrieved.

Top

---

## Examples

```
RTVSMGOBJ OBJ(DEVLLIB/NEWPGM) OBJTYPE(*PGM)
          CPNAME(*NETATR DEVSYS)
```

This command retrieves program NEWPGM in library DEVLLIB from the development system.

Top

---

## Error messages

None

[Top](#)

## Run SMG Command (RUNSMGCMD)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Run System Manager Command (RUNSMGCMD) command provides the capability of sending a command from the central site system to one or more managed systems.

### Restriction:

If a node list (NODL) value is specified, the node list can only contain entries that have a value of \*SNA for the address type.

### Note:

A change request is automatically submitted that can be viewed to determine the status of this command. A message is returned identifying the name of the change request.

Top

## Parameters

Keyword	Description	Choices	Notes
CMD	Command to run	<i>Command string</i>	Required, Positional 1
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: *NONE Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CPNAME	Managed system node names	Single values: *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Communications name</i>	
RMTSTRTIME	Start time on managed system	<i>Element list</i>	Optional
	Element 1: Time zone	*LCLSYS, *MGDSYS	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time, *CURRENT</i>	
	Element 2: Date	<i>Date, *CURRENT, *NEXT</i>	
	Element 3: Start before	<i>Element list</i>	
	Element 1: Time	<i>Time, *ANY, *CURRENT</i>	
	Element 2: Date	<i>Date, *ANY, *CURRENT, *NEXT</i>	
RTNSPLF	Return spooled file	*YES, *NO, *FAIL	Optional
USRPRF	User profile	<i>Simple name, *NONE</i>	Optional
PASSWORD	Password	<i>Character value, *USRPRF, *NONE</i>	Optional
ENCODE	Encode command	*YES, *NO	Optional

---

## Command to run (CMD)

Specifies the command to run. The command can be any command that is run in batch mode. The command must follow the OS/400 CL command format of 1 to 10 characters with the first character alphabetic and the other nine characters alphanumeric. You can use F4 to enable prompting for a CL command.

Top

---

## Managed systems node list (NODL)

Specifies the node list object name that contains a list of systems which are the destinations for the request. It cannot be specified if control point name (CPNAME) parameter is also specified.

The possible value is:

### **\*NONE**

The systems on which this activity is to be performed are not identified by a node list. Individual control point names must be specified.

The possible values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

### **\*CURLIB**

The current library for the job is used to locate the node list object.

### *library-name*

Specify that only the library named in this parameter is searched.

The possible node list name value is:

### *node-list-name*

Specify the node list object name which contains the list of systems on which the command is to be performed.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if the node list (NODL) parameter is specified.

The possible value is:

### **\*NONE**

The systems on which this activity is to be performed are not identified individually. A node list must be specified.

The possible values are:

### **\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.



*network-identifier*

The APPN network identifier of the managed system on which the activity is to be performed.

The possible control point name value is:

*control-point-name*

The APPN control point name of the managed system on which the activity is to be performed.

Top

---

## Start time on managed system (RMTSTARTIME)

Specifies the date and time when the activity can begin running on the managed system. The current date and time values and next date values are determined when the activity begins running at the central site system based on the central site date and time.

The possible time zone values are:

The time zone of the remote start time.

**\*LCLSYS**

The remote start time is specified in the time zone of the central site system.

**\*MGDSYS**

The remote start time is specified in the time zone of the managed system.

The possible start after time values are:

This is the definition of the time after which the activity is to start.

**\*CURRENT**

This function can start on the managed system at any time on or after the time this activity is started on the central site system on the date specified in element 3.

*start-after-time*

Specify the time when this function can start on the managed system. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start after date values are:

**\*CURRENT**

This function starts on the managed system on any date on or after the activity starts on the central site system.

**\*NEXT**

This function starts on the managed system on any date after the activity starts on the central site system.

*start-after-date*

Specify the date after the functions start on the managed system. The date must be specified in the job date format.

The possible start before time values are:

This element is ignored if start before date is \*ANY.

**\*ANY** The activity can start at any time on or after the start time.

#### **\*CURRENT**

The activity must start before the time when the change request is submitted on the date specified on the start before date element.

#### ***start-before-time***

Specify the time before which the activity must be started. If the activity cannot start before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

The possible start before date values are:

**\*ANY** The activity can start at any date on or after the start date.

#### **\*CURRENT**

The activity must start on the date the change request was submitted.

#### **\*NEXT**

The activity must start by the day after the date the change request was submitted.

#### ***start-before-date***

Specify the date before which the activity must be started. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

#### **Note:**

The special values \*CURRENT and \*NEXT cannot be specified for the date and time if the time zone value \*MGDSYS is specified.

Top

---

## **Return spooled file (RTNSPLF)**

Specifies that the output spooled files from the remote command that was submitted returns.

The possible values are:

**\*YES** All spooled files created from the remote command is returned from the managed system. The spooled files from all the nodes are combined into one spooled file that can be viewed by displaying the command activity details of the change request.

**\*NO** The output data is not returned from the managed system.

**\*FAIL** The spooled file job log is returned from the managed system if the command fails when running.

Top

---

## **User profile (USRPRF)**

Specifies the user profile under which the command runs at the remote system. If the remote system is running, the NetView Remote Operations Agent/400 product this parameter is ignored at the remote system.

The possible values are:

#### **\*NONE**

No user profile is specified. The default user profile is used on each remote system.

### *user-profile*

Specify the name of the user profile.

Top

---

## Password (PASSWORD)

The password for the remote user profile.

- Managed Systems at releases prior to V5R1M0 only accept uppercased passwords up to 10 characters long. If a longer password is entered, SMU18A2 message with 100B0007 SNA sense code is returned, indicating that the request was rejected.
- Managed Systems at release V5R1M0 and above, running with QPWDLVL system value:
  - 0 or 1 — truncate the received passwords to 10 characters.
  - 2 or 3 — accept passwords up to 128 characters.

The possible values are:

### **\*NONE**

No password is specified.

### **\*USRPRF**

The password is the same as the user profile.

### *password*

Specify the password for the user profile.

Top

---

## Encode command (ENCODE)

The password for the remote user profile.

The possible values are:

**\*NO** The command, user profile, and password are not encoded when the request is sent to the remote system.

**\*YES** The command, user profile, and password are encoded when the request is sent to the managed systems. The remote command key managed system attribute must be specified on both the central site system and the managed system. This attribute can be changed using the Change Managed System Attributes (CHGMGDSYSA) command. The remote command key must be the same on the central site system and the managed system.

Top

---

## Examples

```
RUNSMGCMD  CMD(STRSBS QSNADS)  NODL(MYLIB/SALESOFC)
            RTNSPLF(*FAIL)
```

This command sends a command to start subsystem QSNADS to the systems identified in node list SALESOFC in library MYLIB. Job log spooled files will be returned if the STRSBS command fails.

Top

---

## Error messages

None

[Top](#)

# Run SMG Object (RUNSMGOBJ)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Run System Manager Object (RUNSMGOBJ) command provides the capability of running an object or sending a command from the central site system to one or more managed systems.

## Note:

A change request is automatically submitted that can be viewed to determine the status of this command. A message is returned identifying the name of the change request.

## Restrictions:

1. Only OS/400 program objects, or file members such as \*CL, \*REXX, and \*UNSPEC can be run.
2. The object cannot reside in the QTEMP library.
3. If a node list (NODL) value is specified, the node list can only contain entries that have a value of \*SNA for the address type.

Top

## Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Object	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *CURLIB</i>	
OBJTYPE	Object type	*FILE, *FILEDATA, *PGM	Required, Positional 2
MBR	Member	<i>Name</i>	Optional
DATATYPE	Data type	*UNSPEC, *CL, *REXX	Optional
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: *NONE Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CPNAME	Managed system node names	Single values: *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Communications name</i>	
PARM	Parameters	Values (up to 20 repetitions): <i>Not restricted</i>	Optional

Top

---

## Object (OBJ)

Specifies the qualified program to be run.

The possible library values are one of the following:

### **\*CURLIB**

The current library for the job is used to locate the object.

### *library-name*

Specify that only the library named in this parameter is searched.

The possible object name value is:

### *object-name*

Specify that only characters A through Z, 0 through 9, dollar sign, at sign, and number sign are used in the object name.

Top

---

## Object type (OBJTYPE)

Specifies the object to be used.

**\*PGM** When this is selected, a program object is to be run.

### **\*FILEDATA**

A CL input stream or a REXX procedure contained in a file is to be run.

**\*FILE** The name of the containing the member with the CL input stream to be run.

Top

---

## Member (MBR)

Specifies the OS/400 physical file member. It is ignored unless the object type is \*FILE. (Example: OS/400 CL input stream.)

### *member-name*

Specify the name of the member to be run.

Top

---

## Data type (DATATYPE)

Specifies the data type of the member. This parameter specifies the type of source file that runs on the managed system.

### **\*UNSPEC**

Unspecified file member type. If the data type cannot be determined at the managed system, or if the name of the file where this member resides is QCLSRC, then the file member is treated as a CL batch input stream. If the source file is named QREXSRC, the file member is treated as a REXX procedure.

**\*CL** The file member contains control language in other words, a CL batch input stream.

### **\*REXX**

The file member contains a REXX procedure.

---

## Managed systems node list (NODL)

Specifies the node list name that contains a list of systems which are the destinations for the activity. This parameter cannot be specified if the control point name (CPNAME) parameter is also specified.

The possible values are:

**\*NONE**

The systems on which this activity is to be performed are not identified by a node list. Individual control point names must be specified.

The possible library values are one of the following:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

**\*CURLIB**

The current library for the job is used to locate the node list object.

*library-name*

Specify that only the library named in the parameter is searched.

The possible node list name value is:

*node-list-name*

Specify the node list object name that contains the list of systems on which the activity is to be performed.

---

## Managed system node names (CPNAME)

Specifies the managed systems APPN control point names on which this activity is performed. Control point names cannot be specified if a node list is specified.

**\*NONE**

The systems on which this activity is performed are not identified individually. A node list must be specified.

**\*NETATR**

The local system network ID is used. This is useful when the node specified is in the same network as the local system.

*network-identifier*

Specify the managed system APPN network identifier on which the activity is performed.

The possible control point name value is:

*control-point-name*

Specify the managed system APPN control point name on which the activity is performed.

---

## Parameters (PARM)

Specifies that the parameters to be passed when starting the program. Up to 20 parameters can be specified.

The possible values are:

### *parameter*

Specify a 1 to 253 character parameter. The prompt panel initially allows 50 characters to be entered. By entering an ampersand (&) in position 1, the field expands for larger parameters.

Top

---

## Examples

### Example 1: Running a REXX Procedure

```
RUNSMGOBJ OBJ(DEVLIB/QREXSRC) OBJTYPE(*FILEDATA)
          MBR(REPORT1) CPNAME((*NETATR DEVSYS))
```

This command runs the REXX procedure in member REPORT1 of file QREXSRC in library DEVLIB on the development system.

### Example 2: Running a Program

```
RUNSMGOBJ OBJ(SALESLIB/GENRPTS) OBJTYPE(*PGM)
          NODL(SYSLIB/ALLSTORES)
          PARM('SALES ONLY' 'TODAY')
```

This command runs a program called GENRPTS in library SALELIB on all systems listed in the node list ALLSTORES in library SYSLIB. Two parameters are passed as input to the program. The first parameter has the value 'SALES ONLY' and the second parameter has the value 'TODAY'.

Top

---

## Error messages

None

Top



---

# Submit Change Request (SBMCRQ)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Submit Change Request (SBMCRQ) command submits a change request to be run.

## Restriction:

You must have \*USE authority to the change request description object the submitted change request is based on.

## Notes:

The following notes provide information on how the command works.

1. If the USRPRF attribute of the change request description specified is \*SBM, this submitted change request runs under the profile that is running the SBMCRQ command. If the USRPRF attribute of the change request description is \*OWNER, this submitted change request runs under the user profile of the owner of the change request description. For more information on the user profile attribute, see the USRPRF parameter of the Create Change Request Description (CRTCRQD) command and of the Change Request Description (CHGCRQD) command in the CL topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.
2. If a node list object is referred in the change request and the node list changes after the change request submission, the submitted change request is not effected. If the change request description of the submitted change request is deleted after the submission is complete, the submitted change request is not effected.
3. If the System Manager licensed program is not active, an information message is sent to the user of the SBMCRQ command. The message indicates that the submitted change request is not processed until the System Manager licensed program starts.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Change request description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CRQ	Change request	<i>Element list</i>	Optional
	Element 1: Name	<i>Name, *CRQD</i>	
	Element 2: Sequence number	<i>1-999999, *GEN</i>	

Top

---

## Change request description (CRQD)

Specifies the qualified name of the change request description object.

The possible library values are:

**\*LIBL** Locates the change request description in the library list of the current job.

**\*CURLIB**

Locates the change request description in the current library. If no library is specified as the job's current library, the QGPL library is used.

*library-name*

Search only the library specified in this parameter.

The possible change request description name value is:

*change-request-description-name*

Specify the name of the change request description on which the change request to submit is based.

Top

---

## Change request (CRQ)

Specifies the change request name and the sequence number of the submitted change request. The sequence number is a 6-digit number that uniquely defines the change request.

The possible values are:

**\*CRQD**

Use the same name as the change request description specified.

*change-request-name*

Specify the name of the change request to submit.

The possible values are:

**\*GEN**

Generates a sequence number. The initial sequence number is 000010. Subsequent sequence numbers are generated in increments of ten. If the sequence number generated is already in use, the next increment of ten is used.

*sequence-number*

Specify the sequence number. The valid values range from 1 to 999999.

Top

---

## Examples

### Example 1: Submitting a Change Request

```
SBMCRQ CRQD(MYLIB/CHG222) CRQ(*CRQD 123)
```

This command submits a change request based on the change request description CHG222 in MYLIB. The change request name submitted is CHG222 with a sequence number of 123.

### Example 2: Submitting a Change Request

```
SBMCRQ CRQD(*CURLIB/CHG999) CRQ(CRQ0001 *GEN)
```

This command shows how to submit change request CRQ0001 with sequence number 00010 if it is not already being used. The submitted change request is based on change request description CHG999 in the current library.

[Top](#)

---

## Error messages

None

[Top](#)



---

## Send License (SNDLIC)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Send License (SNDLIC) command provides the capability to send one or more license keys from the central site system to one or more managed systems. Only the license keys matching the system serial number will be added to the managed system license repository.

### Notes:

1. A change request is automatically submitted which can be viewed to determine the status of this command. A message is returned identifying the name of the change request.
2. All the existing records in the central site license repository containing the license key information for the specified product will be sent to the specified managed system or systems. Only those ones that match the system serial number are added to the managed system license repository.

### Restrictions:

1. This command is shipped with public authority of \*EXCLUDE.
2. If a node list (NODL) value is specified, the node list can only contain entries that have a value of \*SNA for the address type.
3. The license key information for the specified product and managed system must exist in the central site license repository.

Top

---

## Parameters

Keyword	Description	Choices	Notes
PRDID	Product ID	Character value, *ALL	Required, Positional 1
LICTRM	License term	Character value, *ALL	Required, Positional 2
NODL	Managed systems node list	Element list	Optional
	Element 1: Node list	Single values: *NONE Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
CPNAME	Managed system node names	Single values: *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	Communications name, *NETATR	
	Element 2: Control point	Communications name	

Top

---

## Product ID (PRDID)

Specifies the 7-character identifier of the product for which the license key information will be distributed.

The possible values are:

**\*ALL** The license keys for all the products existing in the central site license repository will be distributed.

### *product-ID*

Specifies the 7-character identifier of the product for which the license key information will be distributed.

Top

---

## License term (LICTRM)

Specifies the license term for which the licenses key information will be sent.

The possible values are:

**\*ALL** The product licenses for all license term found will be sent.

### *license-term*

Specify the license term in one of the following formats:

**Vx** The authorized usage limit is valid for the entire version of the product or feature.

**VxRy** The authorized usage limit is valid for the entire release of the product or feature.

### **VxRyMz**

The authorized usage limit is valid only for a modification of the product.

where x and y can be a number from 0 through 9, and z can be a number 0 through 9 or a letter A through Z.

Top

---

## Managed systems node list (NODL)

Node list object name that contains a list of systems which are the destinations for the request. Cannot be specified if control point names (CPNAME parameter) are also specified.

The possible values are:

### **\*NONE**

The systems on which this activity is to be performed are not identified by a node list. Individual control point names must be specified.

The possible library values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

### **\*CURLIB**

The current library for the job is used to locate the node list object.

*library-name*

Only the library named in this keyword is searched.

The possible value for the node list name is:

*node-list-name*

Node list object name which contains the list of systems on which the command is to be performed.

Top

---

## Managed system node names (CPNAME)

The APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if the node list (NODL) parameter is specified.

\*NONE

The systems on which this activity is to be performed are not identified individually. A node list must be specified.

\*NETATR

The network ID of the local system will be used. This useful when the node being specified is in the same network as the local system.

*network-identifier*

The APPN network identifier of the managed system on which the activity is to be performed, it is a character field of length 8.

*control-point-name*

The APPN control point name of the managed system on which the activity is to be performed, it is a character field of length 8.

Top

---

## Examples

```
SNDLIC PRDID(1ACCOUN) LICTRM(V4) CPNAME((*NETATR SYS1))
```

Send the licenses for 1ACCOUN product with license term V4, and add the licenses matching the system serial number to the repository of system SYS1.

Top

---

## Error messages

None

Top





---

## Send Product (SNDPRD)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Send Product (SNDPRD) command provides the capability to send a product, packaged with the iSeries product packaging support, from the central site system to one or more managed systems.

### Note:

A change request is automatically submitted which can be viewed to determine the status of this command. A message is returned identifying the name of the change request.

### Restrictions:

1. This command is shipped with public authority of \*EXCLUDE.
2. Because this command calls the PKGPRDDST and ADDPRDCRQA commands, you must be authorized to use these commands. You must have \*ALL authority to folders.
3. If a NODL value is specified, the node list can only contain entries that have a value of \*SNA for the address type.
4. The Managed System Services licensed program and the OS/400 product cannot be specified.

### Notes:

1. This command packages the product for distribution if it is not already packaged.
2. If the product was packaged for distribution before but the PKGPRDDST(\*REPLACE) parameter was specified, this command packages the product for distribution again. It replaces the previous packaged product.
3. When it is requested to send the license key of the product, a record with the license key information must exist in the central site license repository by the time this activity runs.
4. All the existing records in the central site license repository containing the license key information for the specified product will be sent to the specified managed system or systems, but only those that match the system serial number are added to the managed system license repository.

Top

---

## Parameters

Keyword	Description	Choices	Notes
PRDID	Product ID	<i>Character value</i>	Required, Positional 1
RLS	Release level	<i>Character value, *ONLY</i>	Optional
OPTION	Product option	1-99, * <u>BASE</u>	Optional
LODTYPE	Product load type	* <u>ALL</u> , *CODE, *LNG	Optional
LODID	Load ID	2900-2999, * <u>ALL</u> , *CODE	Optional
TGTRLS	Target release	* <u>CURRENT</u> , *PRV, V5R1M0, V5R2M0, V5R3M0	Optional

Keyword	Description	Choices	Notes
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: <u>*NONE</u> Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CPNAME	Managed system node names	Single values: <u>*NONE</u> Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Communications name</i>	
INSTALL	Install	<u>*NO</u> , *YES	Optional
KEEPCLGE	Keep catalog entry	<u>*NO</u> , *YES	Optional
SNDLICKEY	Send license key	<u>*NO</u> , *YES	Optional
PKGPRDDST	Package product	<u>*NOREPLACE</u> , *REPLACE	Optional

Top

---

## Product ID (PRDID)

Specifies the 7-character identifier of the product to be sent.

The possible value is:

*product-ID*

Specify the 7-character product ID that is used in the activity.

Top

---

## Release level (RLS)

Specifies which version, release, and modification level of the product is used.

The possible values are:

\*ONLY

Only one version, release, and modification level is used for the product option.

*release-level*

Specify the release level in the format VxRxMy, where Vx is the version number, and My is the modification number. Valid values for x range from 0 through 9. Valid values for y range from 0 through 9 and the letters A through Z.

Top

---

## Product option (OPTION)

Specifies which of the optional parts of the product given in the PRDID parameter are used.

The possible values are:

\*BASE

Only the base part of the product is used.

*product-option-number*

Specify the option number for the product load being saved. Valid values are from 1 through 99.

Top

---

## Product load type (LODTYPE)

Specifies the product load objects being used.

The possible values are:

**\*ALL** Code and language objects specified on the LODID parameter are used.

**\*CODE**

The program objects associated with this product load are used.

**\*LNG** The objects associated with the NLV identified on the LODID parameter are used.

Top

---

## Load ID (LODID)

Specifies the load identifier used.

The possible values are:

**\*ALL** All languages for this product option are saved.

**\*CODE**

The code load is to be used.

*product-load-ID*

Specify the load ID of the product when LODTYPE(\*LNG) or LODTYPE(\*ALL) is used. The load ID must be one of the valid IBM national language versions and be specified in the form 29xx. The value of x can be from 0 through 9.

Top

---

## Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the product.

**\*CURRENT**

The product is used on the release of the operating system currently running on your system.

**\*PRV** The product is to be used on the previous release.

*release-level*

Specify the release level in the VxRxMx format. The product can be used on a system with the specified release or with any later release of the operating system installed.

Valid values depend on the current version, release, and modification level, and they change with each new release.

Top

---

## Managed systems node list (NODL)

Specify the node list object name that contains a list of systems which are the destinations for the request. Node list name cannot be specified if a value is specified for the control point name (CPNAME) parameter.

The possible value is:

**\*NONE**

The systems on which this activity is to be performed are not identified by a node list. Individual control point names must be specified.

The possible library values are:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

**\*CURLIB**

The current library for the job is used to locate the node list object.

*library-name*

Specify that only the library named in this parameter is searched.

The possible node list name value is:

*node-list-name*

Specify the node list object name which contains the list of systems on which the command is to be performed.

Top

---

## Managed system node names (CPNAME)

Specifies the APPN control point names of the managed systems on which this request is to be performed. Control point names cannot be specified if a node list is specified.

The possible value is:

**\*NONE**

The systems on which this activity is to be performed are not identified individually. A node list must be specified.

The possible values are:

**\*NETATR**

The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

*network-identifier*

Specify the APPN network identifier of the managed system on which the activity is to be performed. It is an 8-character field.

The possible control point name value is:

*control-point-name*

Specify the APPN control point name of the managed system on which the activity is to be performed.

Top

---

## Install (INSTALL)

Specifies that the product is installed on the managed system when it arrives.

The possible values are:

\*NO The product is not installed.

\*YES The product is installed.

Top

---

## Keep catalog entry (KEEPCLGE)

Specifies if the catalog entry and associated save file corresponding to the product is kept on the specified system or systems. This is only valid if INSTALL(\*YES) was specified.

The possible values are:

\*NO The catalog entry and associated save file are not kept.

\*YES The catalog entry and associated save file are kept.

Top

---

## Send license key (SNDLICKEY)

Specifies if the license key is to be sent with the product.

The possible values are:

\*YES The license key is sent with the product.

\*NO The license key is not sent with the product.

Top

---

## Package product (PKGPRDDST)

Specifies that the product is packaged for distribution before it is sent. This parameter specifies whether to replace the packaged product if it already exists in the distribution repository.

The possible values are:

\*NOREPLACE

Specify that the existing packaged product is not replaced.

\*REPLACE

Specify that the existing packaged product is replaced.

Top

---

## Examples

### Example 1: Sending a Product

```
SNDPRD PRDID(1ACCOUN) RLS(V5R2M0) OPTION(*BASE)
        LODTYPE(*ALL) LODID(*ALL) CPNAME((*NETATR SYS1))
        INSTALL(*YES) SNDLICKEY(*YES)
```

This command sends the 1ACCOUN product base option release V5R2M0 for both the program and the language parts, and install it on the iSeries server SYS1. The license key for this product is also sent and is added to the managed system license repository.

**Example 2: Sending but Not Installing an Option**

```
SNDPRD  PRDID(1CHECKS)  RLS(V5R2M0)  OPTION(0010)
        LODTYPE(*ALL)  LODID(*ALL)
        CPNAME((*NETATR SYS2))  SNDLICKEY(*YES)
```

This command sends both the program and the language parts of the 1CHECKS product option 10, release V5R2M0, but does not install the product on the iSeries server SYS2. The license key for this product is also sent and is added to the managed system license repository.

[Top](#)

---

## Error messages

None

[Top](#)

---

## Send Program Temporary Fix (SNDPTF)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Send Program Temporary Fix (SNDPTF) command sends a specific Program Temporary Fix (PTF) and accompanying requisites, or a list of PTFs to a specific requester or list of service requesters. Cover letters can also be sent if they are available. service requester or a list of service requesters.

When using the SNDPTF command, you can request that the PTFs be applied on the service requester when the PTFs arrive on the service requester. When requesting that the PTFs be applied, you can also request an IPL of the service requester's system. The apply PTF and IPL can be scheduled to occur at a later date.

### Notes:

1. A change request is automatically submitted that can be viewed to determine the status of this command. You can use the Work with Submitted Change Request (WRKSBMCRQ) command to monitor the status.
2. A maximum of 300 PTFs can be specified.

### Restrictions:

1. The apply and IPL functions of this command are only supported when the service requester has the Managed System Services licensed program installed.
2. A PTF save file must exist and be released in order to be sent another system.
3. PTFs that are not marked as delayed are applied immediately. PTFs marked as delayed are scheduled to be applied at the next IPL.
4. PTFs marked as delayed can only be applied permanently if they have been previously applied temporarily. Notice that an IPL would have been required to apply them since they are delayed.
5. If a NODL value is specified, the node list must contain only entries that have address types equal to \*SNA.
6. The product, option, and release of the PTF must be supported using the Work with Supported Products (WRKSPTPRD) command.
7. If you are distributing a PTF and the distribution queue is set to \*SNADS in the Work with Service Requesters (WRKSRVRQS) command, you must have \*USE authority to the Send PTF (SNDPTF) command.
8. If you are distributing a PTF and the distribution queue is set to \*SVDS in the Work with Service Requesters (WRKSRVRQS) command, you must have \*USE authority to the Copy PTF (CPYPTF) command.

Top

---

## Parameters

Keyword	Description	Choices	Notes
PTFID	PTF description	Values (up to 300 repetitions): <i>Element list</i>	Required, Positional 1
	Element 1: PTF identifier	<i>Character value</i>	
	Element 2: Product	<i>Character value, *SPT</i>	
	Element 3: Release level	<i>Character value, *ONLY</i>	

Keyword	Description	Choices	Notes
DESTSRVRQS	Destination service requester	Single values: *ALL, *SELECT, *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name</i> , *NETATR	
	Element 2: Control point	<i>Communications name</i>	
MAXDST	Maximum distribution	1-5000, *SRVRQS, *SRVPVDA, *NOMAX	Optional
SNDRQT	Send requisites	*YES, *NO	Optional
CHECK	Check service requesters	*YES, *NO	Optional
PTFPART	PTF parts	*ALL, *PTF, *CVRLTR	Optional
APY	Extent of change	*NONE, *TEMP, *PERM	Optional
DLYAPY	Delayed PTF apply	*NO, *YES	Optional
NODL	Node list name	<i>Qualified object name</i>	Optional
	Qualifier 1: Node list name	<i>Name</i> , *NONE	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
RMTAPYTIME	Remote apply time	<i>Element list</i>	Optional
	Element 1: Time zone	*LCLSYS, *MGDSYS	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *CURRENT	
	Element 2: Date	<i>Date</i> , *CURRENT, *NEXT	
IPL	IPL remote system	*NO, *YES	Optional
RMTIPLTIME	Remote IPL time	<i>Element list</i>	Optional
	Element 1: Time zone	*LCLSYS, *MGDSYS	
	Element 2: Start after	<i>Element list</i>	
	Element 1: Time	<i>Time</i> , *CURRENT	
	Element 2: Date	<i>Date</i> , *CURRENT, *NEXT	
PWRDWNOPT	Power down option	*CNTRLD, *IMMED	Optional
PWRDWNDLY	Power down delay	1-65535, 3600	Optional

Top

## PTF description (PTFID)

Specifies the PTF numbers for the PTFs that are to be sent. The following special PTFs can also be distributed:

### SF97vrm or nnn

PTF summary package

### MF98vrm or nnn

Hardware service planning tips

### SF98vrm or nnn

Software service planning tips

n is the PTF sequence number  
v is the version for the package requested  
r is the release for the package requested  
m is the modification level  
for the package requested

The possible PTF identifier values are:



### *PTF-identifier*

Specify the 7-character identifier (ID) of the PTF that is to be sent.

The possible product identifier values are:

**\*SPT** The PTF identifier is unique to only one supported product.

### *product-id*

Specify the product ID of the PTF that is to be sent. The product must be specified when PTF identifiers are not unique across products.

The possible release level of product values are:

### **\*ONLY**

The PTF identifier is unique to only one release level of the product.

### *release (VxRxMx)*

Specify the version, release, and modification level of the PTF being sent. The release level must be specified when the PTF identifiers are not unique across products. The format is VxRxMx.

Top

---

## **Destination service requester (DESTSRVRQS)**

Specifies the service requester system where this PTF function is performed. A single service requester or a list of service requesters can be specified.

The possible network identifier values are:

### **\*SELECT**

Shows a list of service requesters. From the list, you can select one or more service requesters. \*SELECT is not valid when the command is used in a batch environment.

**\*ALL** Specifies that this PTF function should occur on all of the service requesters that are defined for this service provider.

### **\*NONE**

The service requesters are identified in a node list object.

### **\*NETATR**

The remote network identifier of the service requester is the same as the network identifier of this system.

### *remote-network-identifier*

Specify the remote network identifier of the service requester.

The possible control point value is:

This element is not valid when the network identifier is \*ALL, \*SELECT, or \*NONE.

### *remote-control-point*

Specify the remote control point name of the service requester.

Top

---

## **Maximum distribution (MAXDST)**

Specifies the maximum number of kilobytes that can be sent to each service requester.

The possible values are:

### **\*NOMAX**

Sends any number of kilobytes.

### **\*SRVRQS**

The number of kilobytes that are specified for each service requester in the service requester entry information is used. Use the Work with Service Requesters (WRKSRVRQS) command to display the current value.

### **\*SRVPVDA**

The number of kilobytes specified on the maximum distribution service provider attribute is used. Use the Display Service Provider Attribute (DSPSRVPVDA) command to determine the current value.

### ***maximum-number-of-kilobytes***

Specify the maximum number of kilobytes that can be distributed. Specify a value ranging from 1 through 5000 KB.

Top

---

## **Send requisites (SNDRQT)**

Specifies if requisite PTFs should be added to the list. Requisites includes prerequisite and corequisite PTFs.

The possible values are:

**\*YES** The PTF and all of its requisites are sent.

**\*NO** The specific PTF or list of PTFs are sent. No requisites are sent.

**Note:** When using SNDRQT(\*NO) APY(\*TEMP) or SNDRQT(\*NO) APY(\*PERM) all requisite PTFs must be applied on the service requester. The PTFs are applied in the order listed on this command.

Top

---

## **Check service requesters (CHECK)**

Specifies that each service requester is checked to determine its unique set of required PTFs.

The possible values are:

**\*YES** Each service requester is checked to determine the required set of PTFs.

**\*NO** The service requesters are not checked to determine whether the PTFs are required. All PTFs are sent to all specified service requesters.

Top

---

## **PTF parts (PTFPART)**

Specifies whether PTFs and cover letters should be sent.

The possible values are:

**\*ALL** Sends the PTF and its associated cover letter. If a cover letter is not found for a PTF, the cover letter is not sent.

**\*PTF** Sends only the PTF.

**\*CVRLTR**

Sends only the cover letter.

Top

---

## Extent of change (APY)

Specifies the extent of change when the PTFs are applied. The apply occurs when all the PTFs have been successfully received by the service requester.

The possible values are:

**\*NONE**

Do not apply PTFs on the service requester's system.

**\*TEMP**

Apply the PTFs temporarily on the service requester's system.

**\*PERM**

Apply the PTFs permanently on the service requester's system.

Top

---

## Delayed PTF apply (DLYAPY)

Specifies how PTFs are applied. Apply immediate PTFs at the time the activity runs on a specified system or later during the next IPL.

**\*NO** Applies an immediate PTF at the time the activity runs. If the PTF is marked delayed, it is not applied until the next unattended IPL.

**\*YES** Applies both immediate or delayed PTFs during the next IPL.

Top

---

## Node list name (NODL)

Specifies the node list object name that contains a list of service requesters where this PTF function is to be performed.

The possible values are:

**\*NONE**

The service requesters are identified in the DESTSRVRQS parameter.

The possible values are:

**\*LIBL** Searches all of the libraries in the user and system portions of the job's library list for the node list object.

**\*CURLIB**

Searches for the node list object in the library for the current job.

*library-name*

Specify the name of the library to be searched.

The possible node list name value is:

*node-list-name*

Specify the node list name which contains the list of systems where the PTF function is to be performed.

Top

---

## Remote apply time (RMTAPYTIME)

Specifies the date and time when the apply can occur on the service requester. The current date and time values and next date are determined, when the apply request is processed, based on this system's date and time.

The possible time zone values are:

**\*LCLSYS**

Specifies the remote start time in the time zone of this system.

**\*MGDSYS**

Specifies the remote start time in the time zone of the service requester's system.

The possible start after time values are:

**\*CURRENT**

Starts the apply on the service requester system at any time on or after the apply request is processed on this system.

*start-after-time*

Specify the time when the apply can be started on the service requester. The time can be entered as 4 or 6 digits, such as hhmm or hhmmss where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a separator. With a time separator, specify a string of 5 or 8 digits such as hh:mm:ss.

The possible start after date values are:

**\*CURRENT**

Starts the apply on the service requester system on any date on or after the date the apply request is processed on this system.

**\*NEXT**

Starts the apply on the service requester system on any date after the date the apply request is processed on this system.

*start-after-date*

Specify the date when the apply can start on the service requester system. The date must be specified in the job date format.

**Note:**

The special values \*CURRENT and \*NEXT cannot be specified for the date and time if the time zone value \*MGDSYS is specified.

Top

---

## IPL remote system (IPL)

Specifies if an IPL of the service requester's system should occur.

The possible values are:

**\*NO** An IPL of the service requester system should not occur.

**\*YES** An IPL of the service requester system should occur after all the PTFs are successfully applied on the service requesters system.

Top

---

## Remote IPL time (RMTIPLTIME)

Specifies the date and time when the IPL can occur on the service requester. The current date and time values and next date are determined, when this IPL request is processed on this system, based on this system's date and time.

The possible time zone values are:

### \*LCLSYS

Specifies the remote start time in the time zone of this system.

### \*MGDSYS

Specifies the remote start time in the time zone of the service requester's system.

The possible start after time values are:

### \*CURRENT

Starts the IPL on the service requester system at any time on or after this IPL request is processed on this system.

### *start-after-time*

Specify the time when this IPL can be started on the service requester. The time can be entered as 4 or 6 digits, such as hhmm or hhmmss where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a separator. With a time separator, specify a string of 5 or 8 digits such as hh:mm:ss.

The possible start after date values are:

### \*CURRENT

Starts the IPL on the service requester system on any date on or after the date the IPL request is processed on this system.

### \*NEXT

Starts the IPL on the service requester system on any date after the date this IPL request is processed on this system.

### *start-after-date*

Specify the date when the IPL can start on the service requester system. The date must be specified in the job date format.

**Note:** The special values \*CURRENT and \*NEXT cannot be specified for the date and time if the time zone value \*MGDSYS is specified.

Top

---

## Power down option (PWRDWNOPT)

Specifies how to manage the end of active jobs during the IPL process at the service requester system. This parameter is valid only if IPL(\*YES) is specified.

### \*CNTRLD

All jobs end when the specified time in the Power Down Delay (PWRDWDLY) parameter expires. Any current jobs that are running will perform an end of batch processing.

## \*IMMED

Jobs end immediately.

Top

---

## Power down delay (PWRDWNDLY)

This parameter is valid only if IPL(\*YES) and if PWRDWNOPT(\*CNTRLD) are specified. Specifies the amount of time in seconds that the system allows a controlled end to be performed. If the end of job routines are not finished in the specified power down delay, any remaining jobs are ended immediately.

**3600** The amount of time in which to complete a controlled end of processing is limited to 3600 seconds.

### *delay-time*

Specify the maximum amount of delay time, in seconds, in which a controlled end can be performed. Valid values range from 1 through 65535 (65,535 seconds).

Top

---

## Examples

### Example 1: Sending a PTF and Requisites

```
SNDPTF  PTFID((1234567) (1234600))  DESTSRVRQS(*ALL)
        MAXDST(*SRVRQS)
```

This command sends a PTF and requisites to one or more selected service requesters.

### Example 2: Sending a PTF to a Specific Service Requester

```
SNDPTF  PTFID(1234567)  DESTSRVRQS(*NETATR CHICAGO1)
        SDRQT(*NO)
```

This command distributes PTF 1234567 to a service requester with a control point name of CHICAGO1, if the PTF is not already there. The size of the PTF package does not have an effect on the distribution and no requisites are sent with the PTF.

### Example 3: Sending a PTF and Its Requisites to Two Service Requesters

```
SNDPTF  PTFID(1234567)
        DESTSRVRQS((*NETATR CHICAGO1) (*NETATR CHICAGO2))
        CHECK(*NO)
```

This command distributes PTF 1234567 and its requisites to two service requesters with control point names of CHICAGO1 and CHICAGO2. The service requesters are not checked before the PTFs are distributed to determine whether PTF 1234567 is already there. If any of the PTFs exist on the service requesters, they are replaced.

### Example 4: Selecting One or More Service Requesters as Destination

```
SNDPTF  PTFID(1234567)  DESTSRVRQS(*SELECT)
```

This command shows a display that allows the user to select one or more service requesters as the destination for PTF 1234567 and its requisites.

### Example 5: Sending and Temporarily Applying One PTF

```
SNDPTF  PTFID(1234567 *SPT *ONLY) DESTSRVRQS(*NETATR CHICAGO1)
SNDRQT(*NO) CHECK(*NO) PTFPART(*PTF) APY(*TEMP)
RMTAPYTIME((*MGDSYS ('10:00:00' '10/31/02'))
IPL(*YES)
RMTIPLTIME((*MGDSYS ('01:00:00' '11/01/02'))
```

This command sends and temporarily applies one PTF on a specific requester, schedules when the PTF is applied, and schedules the IPL.

This example loads and applies the specified PTFs at 10:00 a.m. on October 31, 2002 if all the PTFs were received successfully. If the PTFs have been received and applied successfully, the IPL on the remote system starts at 1:00 a.m. on November 1, 2002.

#### Example 6: Sending and Applying a Permanent PTF

```
SNDPTF  PTFID(1234567 *SPT *ONLY)
DESTSRVRQS((*NETATR CHICAGO1) (*NETATR CHICAGO2))
PTFPART(*PTF) APY(*PERM)
RMTAPYTIME((*MGDSYS ('10:00:00' '10/31/02'))
IPL(*YES)
RMTIPLTIME((*MGDSYS ('01:00:00' '11/01/02'))
```

This command sends and permanently applies one PTF to more than one service requester. This schedules the application of the PTF and the IPL of the system. The scheduled time is the time at the location of the service requester or the remote system. The command sends requisites and checks the service requester for the availability of the PTFs.

This example applies the specified PTFs on the service requester starting at 10:00 a.m. on October 31, 2002, if all the PTFs were received successfully. If the PTFs have been received and applied successfully, the remote system IPL starts at 1:00 a.m. on November 1, 2002.

#### Example 7: Doing an IPL after Sending and Applying a PTF

```
SNDPTF  PTFID(1234567 *SPT *ONLY) DESTSRVRQS(*SELECT)
PTFPART(*PTF) APY(*TEMP)
RMTAPYTIME((*MGDSYS (*CURRENT *CURRENT)) IPL(*YES)
RMTIPLTIME((*LCLSYS (*CURRENT 12/24/02))
PWRDWNOPT(*CNTRL) PWRDWNPLY(4800)
```

This command sends and temporarily applies one PTF to one or more selected service requesters. This command schedules the application of the PTF and the IPL of the system. The scheduled time is the time at the location of the service requester or the remote system. The command sends requisites and checks the service requester or the availability of the PTFs.

This example applies the specified PTF on the service requester starting at the current time on the current date. If the PTFs were received and applied successfully, the remote system IPL starts at the current time on December 24, 2002. The time specified for the IPL is in the time zone of the local system.

Top

---

## Error messages

### \*ESCAPE Messages

#### SMU107F

Error found processing &1 command.

#### SMU1080

Cannot specify \*SELECT for service requester destination.

**SMU109C**

Errors detected while processing PTFs.

**SMU109E**

One or more PTFs could not be sent.

**SMU109F**

No PTFs sent.

**\*STATUS Messages****SMU109A**

Contacting &1.&2 to determine required PTFs.

[Top](#)



## Send SMG Object (SNDSMGOBJ)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Send System Manager Object (SNDSMGOBJ) command provides the capability of sending an OS/400 object from the central site system to one or more managed systems.

**Note:** A change request is submitted that can be viewed to determine the status of this command. A message is returned identifying the name of the change request.

### Restrictions:

1. Only OS/400 program objects or file members such as \*REXX, \*CL, and \*UNSPEC can be run.
2. The object cannot reside in the QTEMP library.
3. If a node list (NODL) value is specified, the node list can only contain entries that have a value of \*SNA for the address type.

## Error messages for SNDSMGOBJ

None

Top

## Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Object	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OBJTYPE	Object type	<i>Character value</i>	Required, Positional 2
MBR	Member	<i>Name, *ALL, *FIRST, *LAST</i>	Optional
DATATYPE	Data type	<i>*UNSPEC, *CL, *REXX</i>	Optional
NODL	Managed systems node list	<i>Element list</i>	Optional
	Element 1: Node list	Single values: <b>*NONE</b> Other values: <i>Qualified object name</i>	
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CPNAME	Managed system node names	Single values: <b>*NONE</b> Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Communications name</i>	
TGTRLS	Target release	<i>*CURRENT, *PRV, V5R1M0, V5R2M0, V5R3M0</i>	Optional
REPLACE	Replace object	<i>*NO, *YES</i>	Optional
DTACPR	Data compression	<i>*NONE, *SNA</i>	Optional
RUNPGM	Run program	<i>*NO, *YES</i>	Optional

Keyword	Description	Choices	Notes
PARM	Parameters	Values (up to 20 repetitions): <i>Not restricted</i>	Optional

Top

---

## Object (OBJ)

Specifies the qualified name to be sent.

The possible library values are one of the following:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched.

**\*CURLIB**

The current library for the job is used to locate the object.

*library-name*

Specify only the library named in this parameter is searched.

The possible object name value is:

*object-name*

Specify that characters A through Z, 0 through 9, dollar sign, at sign, and number sign are used in the object names.

Top

---

## Object type (OBJTYPE)

Specifies the object.

**\*FILEDATA**

A file member should be sent without the file attributes.

*object-type*

Specify the object type.

Top

---

## Member (MBR)

Specifies the OS/400 physical file member name. It is ignored unless the object type is \*FILE or \*FILEDATA.

**\*ALL** The entire file is sent. \*ALL must be used if the file is not a physical file. \*ALL cannot be used if RUNPGM(\*YES) is selected or if \*FILEDATA is selected.

**\*FIRST**

This is used only if the first member is sent. The member name is determined when the activity is run.

**\*LAST**

Only the last member is sent. The member name is determined when the activity is run by the date the member was added to the physical file.

*member-name*

Specify the name of the member to be sent.

---

## Data type (DATATYPE)

Specifies the data type of the member. This parameter needs to be specified only when sending a source file that runs on the managed system. DATATYPE is specified only when running \*FILEDATA.

### \*UNSPEC

Unspecified file member type. If the data type cannot be determined at the managed system, or if the name of the file where this member resides is QCLSRC, then the file member is treated as a CL batch input stream. If the source file is named QREXSRC, the file member is treated as a REXX procedure.

**\*CL** The file member contains control language such as the OS/400 CL batch input stream.

### **\*REXX**

Specifies the file member contains a REXX procedure.

---

## Managed systems node list (NODL)

Specifies the node list name contains a list of systems which are the destinations for the activities. This parameter cannot be specified if control point name (CPNAME) parameter is also specified.

The possible values are:

### \*NONE

The systems on which this activity is performed are not identified by a node list. Individual control point names must be specified.

The possible library values are one of the following:

**\*LIBL** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

### **\*CURLIB**

The current library for the job is used to locate the node list object.

### *library-name*

Specify that only the library named in the parameter is searched.

The possible node list name value is:

### *node-list-name*

Specify the node list object name containing the list of systems on which the activity is to be performed.

---

## Managed system node names (CPNAME)

Specifies the managed systems APPN control point names on which this request is performed. Control point names cannot be specified if a node list is specified.

### \*NONE

The systems on which this request is performed are not identified individually. A node list must be specified.

### **\*NETATR**

The local system network ID is used. This is useful when the node specified is in the same network as the local system.

#### *network-identifier*

Specify the managed system APPN network identifier on which the request is performed.

The possible control point name value is:

#### *control-point-name*

Specify the managed system APPN control point name on which the request is performed.

Top

---

## **Target release (TGTRLS)**

Specifies the release of the operating system on which you intend to use the object. This parameter is ignored for objects with global names that are in the SystemView distribution repository or for actions other than send or retrieve.

### **\*CURRENT**

The object is used on the release of the operating system currently running on your system. If V5R3M0 is running on your system, \*CURRENT means that you intend to use the object on a system with V5R3M0 installed. The object can also be used on a system with any later release of the operating system installed.

**\*PRV** The object is intended for a system which is at the previous release level compared to the local system.

#### *release-level*

Specify the release level in the VxRxMx format. The object is used on a system with the specified release or with any later release of the operating system installed.

Valid values depend on the current version, release, and modification level, and they change with each new release.

Top

---

## **Replace object (REPLACE)**

Specifies that the object should be replaced if it already exists.

The possible values are:

**\*NO** An error is returned if the object already exists.

**\*YES** The object is replaced if it already exists.

Top

---

## **Data compression (DTACPR)**

Specifies that the data is compressed when sent. This parameter is valid only if the object type \*FILEDATA is specified. SNA compression character or blank is performed.

### **\*NONE**

The file data is not compressed when sent.

**\*SNA** The file data is compressed when sent.

---

## Run program (RUNPGM)

Specifies running the program when it arrives at the managed system. This parameter can only be specified if the object type is \*PGM or if the object type is \*FILEDATA and the data type is \*CL or \*REXX.

**\*NO** The program is not run.

**\*YES** Specify that the program is run as soon as it is received by the managed system.

---

## Parameters (PARM)

Specifies the parameters to be passed when starting the program. Up to 20 parameters can be specified.

### *parameter*

Specify a 1 to 253 character parameter. The prompt panel initially allows 50 characters to be entered. By entering an ampersand (&) in position 1, the field expands for larger parameters.

---

## Examples

### Example 1: Sending a Library to Systems in a Node List

```
SNDSMGOBJ OBJ(ACCTLIB) OBJTYPE(*LIB) NODL(SYSLIB/SALESOFC)
```

This command sends library ACCTLIB to 20 systems identified in node list SALESOFC.

### Example 2: Sending a Program to All Systems in the Node List

```
SNDSMGOBJ OBJ(SALESLIB/GENRPTS) OBJTYPE(*PGM)
          NODL(SYSLIB/ALLSTORES)
          RUNPGM(*YES) PARM(SALES 'ONE DAY')
```

This command sends the program GENRPTS in library SYSLIB to all the systems listed in node list ALLSTORES in library SYSLIB. It then runs the program on each of those systems. Two parameters are passed as input to the program. The first has the value SALES and the second has the value 'ONE DAY'.

---

## Error messages

None



---

# Start System Manager (STRSYSMGR)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

[Parameters](#)  
[Examples](#)  
[Error messages](#)

The Start System Manager (STRSYSMGR) command starts the job in the QSYSWRK subsystem that run the System Manager licensed program functions.

The Start Managed System (STRMGDSYS) command is used to start the services that are common to both a central site system and a managed system. These services must be running in order for the central site system support to be fully operational.

The jobs running the managed system functions, which are started by the STRMGDSYS command, are checked and a diagnostic message is sent if they are not started. This message is only sent for managed system functions that apply to a central site system.

## Restrictions:

1. You must have \*JOBCTL authority to use this command.
2. \*PUBLIC authority for this command is \*EXCLUDE.

[Top](#)

---

## Parameters

None

[Top](#)

---

## Examples

STRSYSMGR

This command starts the jobs in the QSYSWRK subsystem that run the System Manager functions.

[Top](#)

---

## Error messages

None

[Top](#)





---

## Work with CRQ Descriptions (WRKCRQD)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Work with Change Request Descriptions (WRKCRQD) command shows the Work with Change Request Descriptions display. You can create, change, and delete change request descriptions, and work with change request description activities using this command.

### Restrictions:

1. Searches only the libraries to which you have \*USE authority.
2. Shows only the change request descriptions to which you have authority.
3. To perform operations on the change request descriptions, you must have \*USE authority to the command used by the operation and the appropriate authority to the change request descriptions on which the operation is performed.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CRQD	Change request description	<i>Qualified object name</i>	Optional, Positional 1
	Qualifier 1: Change request description	<i>Generic name, name, <u>*ALL</u></i>	
	Qualifier 2: Library	<i>Name, <u>*LIBL</u>, *CURLIB, *USRLIBL, *ALL, *ALLUSR</i>	

Top

---

## Change request description (CRQD)

Specifies the qualified name of the change request description object.

The possible change request description values are:

**\*ALL** Display all change request descriptions.

### *change-request-description*

Specify the change request description to display.

### *generic\*-change-request-description*

Specify a generic change request description name. A generic name is a character string containing one or more characters followed by an asterisk (\*). Only the change request descriptions with a matching name or with a matching beginning string display.

The possible library values are:

**\*LIBL** Searches all libraries in your library list and in the system portion of the job's library list.

### **\*CURLIB**

Locate the change request description on the current library. If no library is specified as the current library for the job, the QGPL library is used.

**\*USRLIBL**

A problem log entry is created when a remote alert is received.

**\*ALL** Search all libraries in the system including the QSYS library.

**\*ALLUSR**

Search all non-system libraries including all user-defined libraries and the QGPL library. Libraries starting with the letter Q other than QGPL are not included.

*library-name*

Specify the library to be searched in this parameter.

Top

---

## Examples

### Example 1: Working with Change Request Descriptions

```
WRKCRQD
```

This command shows how to work with all change request descriptions in the library list of the job.

### Example 2: Working with All Change Request Descriptions

```
WRKCRQD CRQD(MYLIB/*ALL)
```

This command shows how to work with all change request descriptions in MYLIB.

### Example 3: Working with All Change Request Descriptions in a String

```
WRKCRQD CRQD(CHG*)
```

This command shows how to work with all change request descriptions that begin with the letters 'CHG' and are in the library list of the current job.

Top

---

## Error messages

None

Top

---

## Work with PTF (WRKPTF)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Work with PTF (WRKPTF) command selects and works with all program temporary fixes (PTFs) for a product.

Top

---

### Parameters

Keyword	Description	Choices	Notes
PTF	PTF identifier	Character value, <u>*ALL</u> , *PTFSAVE, *ONORDER, *CVLT, *CVLTONLY	Optional, Positional 1
LICPGM	Product	Character value, <u>*SELECT</u>	Optional, Positional 2
RLS	Release level	Character value, <u>*ONLY</u>	Optional, Positional 3

Top

---

### PTF identifier (PTF)

Specifies the PTFs with which you want to work.

The possible values are:

\*ALL All PTFs for the specified product and release are shown.

**\*PTFSAVE**

PTFs for the specified product and release that have a save file in QGPL are shown.

**\*ONORDER**

PTFs for the specified product and release that are on order are shown.

**\*CVLTONLY**

PTFs for the specified product and release that contain only a cover letter are shown.

**\*CVLT**

PTFs for the specified product and release that contain a cover letter, regardless of the PTF status, are shown.

**PTF-ID**

Specify the identifier (ID) of the PTF to be shown.

Top

---

## Product (LICPGM)

Specifies the ID of the product for the PTFs being shown.

The possible values are:

### \*SELECT

Shows a list of products from which you can select the PTFs you want to show.

### *product-ID*

Specify the product ID for the PTFs that you want to show.

Top

---

## Release level (RLS)

Specifies the release level of the PTFs that are shown.

The possible values are:

### \*ONLY

Only one release of the specified product is installed or supported on the system.

### *release-level*

Specify the release level in the format VxRyMz, where Vx is the version number, Ry is the release number, and Mz is the modification level. The variables X and y can be a number from 0 through 9, and the variable z can be a number from 0 through 9 or a letter from A through Z.

Top

---

## Examples

```
WRKPTF PTF(*ALL) LICPGM(*SELECT)
```

This command allows you to select a product from a list of products shown.

Top

---

## Error messages

### \*ESCAPE Messages

#### SMU1412

Product &1 release &2 not found.

Top

## Work with Submitted CRQs (WRKSBMCRQ)

Where allowed to run: Interactive environments (\*INTERACT  
 \*IPGM \*IREXX \*EXEC)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Work with Submitted Change Requests (WRKSBMCRQ) command shows the Work with Submitted Change Requests display. From this display, you hold, release, end, and delete submitted change requests and work with submitted change request activities.

### Restrictions:

1. Only the change requests submitted by you are displayed.
2. If you have \*JOBCTL special authority, all change requests are displayed.

Top

## Parameters

Keyword	Description	Choices	Notes
CRQ	Change request	<i>Element list</i>	Optional, Positional 1
	Element 1: Name	<i>Generic name, name, <u>*ALL</u></i>	
	Element 2: Sequence number	1-999999, <u>*ALL</u>	
STATUS	Status	<u>*ALL</u> , *SBM, *ACTIVE, *ENDED	Optional
HIGHENDCDE	Highest end code	0-99, *ANY, *SUCCESS, *FAIL, *NOTRUN	Optional
LASTENDCDE	Last end code	0-99, *ANY, *SUCCESS, *FAIL, *NOTRUN	Optional
CRQD	Change request description	<i>Qualified object name</i>	Optional
	Qualifier 1: Change request description	<i>Generic name, name, <u>*ALL</u></i>	
	Qualifier 2: Library	<i>Name, <u>*ALL</u></i>	
SBM	Submitter	<i>Name, <u>*ALL</u></i>	Optional
PERIOD	Period	<i>Element list</i>	Optional
	Element 1: Start time and date	<i>Element list</i>	
	Element 1: Start time	<i>Time, <u>*AVAIL</u></i>	
	Element 2: Start date	<i>Date, <u>*BEGIN</u>, *CURRENT</i>	
	Element 2: End time and date	<i>Element list</i>	
	Element 1: Ending time	<i>Time, <u>*AVAIL</u></i>	
	Element 2: Ending date	<i>Date, <u>*END</u>, *CURRENT</i>	
PRBID	Problem identifier	<i>Character value, <u>*ALL</u></i>	Optional
PRBORG	Problem origin	<i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, <u>*ALL</u>, *NETATR</i>	
	Element 2: Control point	<i>Communications name, <u>*ALL</u>, *NETATR</i>	

Top

---

## Change request (CRQ)

Specifies the qualified change request name.

The possible change request name values are:

**\*ALL** Display all the qualified change requests.

***change-request-name***

Specify that only the specified change requests display.

***generic\*-change-request-name***

Specify a generic change request name. A generic name is a character string containing one or more characters followed by an asterisk (\*). Only the change request names that match or that match the beginning of the string are displayed.

The possible sequence number values are:

**\*ALL** Display all selected change requests regardless of sequence number.

***sequence-number***

Specify that only the change requests with the specified sequence number are displayed.

Top

---

## Status (STATUS)

Specifies the status for which change request are displayed.

The possible values are:

**\*ALL** Display all qualified change requests regardless of status.

**\*SBM** Display only the change requests with a status of Submitting.

**\*ACTIVE**

Display only the change requests with a status of Active.

**\*ENDED**

Display the change requests with a status of Ending.

Top

---

## Highest end code (HIGHENDCDE)

Specifies the highest end code or range for which to display change requests.

The possible values are:

**\*ANY** Displays change requests regardless of the highest end code.

**\*FAIL** Display all change requests with the highest end code between 10 and 89.

**\*SUCCESS**

Display all change requests with the highest end code between 0 and 9.

**\*NOTRUN**

Display all change requests with the highest end code between 90 and 99.

***highest-end-code***

Specify the highest end code to use. Valid values range from 0 through 99.

---

## Last end code (LASTENDCDE)

Specifies the last change request end code or range for which to display change requests.

The possible values are:

**\*ANY** Display change requests regardless of the last end code.

**\*FAIL** Display all change requests with the last end code between 10 and 89.

**\*SUCCESS**

Display all change requests with the last end code between 0 and 9.

**\*NOTRUN**

Display all change requests with the last end code between 90 and 99.

*last-end-code*

Specify the last end code to use. Valid values range from 0 through 99.

---

## Change request description (CRQD)

Specifies the qualified name of the change request description object.

The possible values are:

**\*ALL** Search all libraries in the system including QSYS.

*library-name*

Specify that the library specified in this parameter be searched.

The possible values for change request description names are:

**\*ALL** Display all change requests regardless of the change request description they are based on.

*change-request-description*

Specify that only change requests based on the specified change request description are displayed.

*generic\*-change-request-description*

Specify a generic change request description name. A generic name is a character string containing one or more characters followed by an asterisk(\*). Display only change requests based on the change request description name matching or matching the beginning of the string.

---

## Submitter (SBM)

Specifies who submitted the change request to be displayed.

**\*ALL** Display all selected change requests regardless of submitter.

*user-profile-name*

Specify that only the change requests submitted by the user profile are displayed.

---

## Period (PERIOD)

Specifies a period of time for which the change requests are displayed. The following values can be coded in this parameter, which contains two list of two elements each:

The possible starting time values are:

One of following specifies the starting time for the change request or the time after the change request must be submitted. Change requests submitted before the specified time and date are not displayed.

### \*AVAIL

The change requests that are available for the specified starting date are displayed.

### *start-time*

Specify the starting time for the specified starting date that indicates the change requests to be displayed.

The possible starting date values are:

One of the following is used to specify the starting date on which or after which the change requests must have been submitted. Entries submitted before the specified date are not displayed.

### \*BEGIN

Change requests are displayed beginning with the date that the first change request was submitted. If \*BEGIN is specified, then any time value other than \*AVAIL for start-time is ignored.

### \*CURRENT

The change requests submitted today between the specified starting and ending times are displayed.

### *start-date*

Specify the starting date for which change requests are displayed.

The possible ending time values are:

One of the following is used to specify the ending time before which the change requests must have been submitted.

### \*AVAIL

The change requests that are available for the specified ending date are displayed.

### *end-time*

Specify the ending time for the specified ending date that determines the change requests to be displayed.

The possible ending date values are:

One of the following is used to specify the ending date before which or on which the change requests must have been submitted.

### \*END

The last day on which change requests have been submitted is the last day for which the change requests are displayed. If \*END is specified, then any value other than \*AVAIL for ending time is ignored.

### \*CURRENT

The change requests submitted today between the specified starting and ending times are displayed.

### *end-date*

Specify the ending date for which change requests are displayed.



---

## Problem identifier (PRBID)

Specifies the problem identifier associated with the change request.

**\*ANY** Display the change requests regardless of the problem identifier.

*problem-identifier*

Specify that the change requests associated with the specified problem identifier be displayed.

Top

---

## Problem origin (PRBORG)

Specifies the originating system of the problem ID.

The possible network ID values are:

**\*ALL** Display change requests regardless of the problem origin network ID.

**\*NETATR**

Display change requests with a problem origin network ID that is the same as the network ID in the system network attributes.

*network-ID*

Specify a problem origin network ID for which change requests are displayed.

The possible control point name values are:

**\*ALL** Display change requests regardless of the problem origin control point name.

**\*NETATR**

Display change requests with a control point name that is the same as the control point name defined in the system network attributes.

*control-point-name*

Specify a problem origin control point name for which change requests are displayed.

Top

---

## Examples

### Example 1: Working with All Change Requests

```
WRKSBMCRQ
```

This command shows how to work with all change requests.

### Example 2: Working with All Active Change Requests

```
WRKSBMCRQ STATUS(*ACTIVE)
```

This command shows how to work with all active change requests.

### Example 3: Working with All Change Requests for a Specific Period

```
WRKSBMCRQ PERIOD((*AVAIL *BEGIN) (*AVAIL *CURRENT))
```

This command shows how to work with all change requests from the beginning of the log to the current date.

#### **Example 4: Working with All Submitted Change Requests**

```
WRKSBMCRQ CRQD(CRQD001)
```

This command shows how to work with all submitted change requests based on the CRQD001 change request description.

[Top](#)

---

## **Error messages**

None

[Top](#)

---

## Work with Submitted CRQ Act (WRKSBMCRQA)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Work with Submitted Change Request Activities (WRKSBMCRQA) command displays the Work with Submitted Change Request Activities display. From this panel, you can display, hold, release, and end activities; display messages and conditions; and work with destination nodes for an activity.

### Restriction:

The activities for the change request specified which satisfy the selection criteria are displayed if you submitted the change request, or you have \*JOBCTL special authority. An error message is sent to you if you try to display activities for a change request that you did not submit or you do not have \*JOBCTL authority.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CRQ	Change request	<i>Element list</i>	Required, Positional 1
	Element 1: Name	<i>Name</i>	
	Element 2: Sequence number	1-999999	
STATUS	Status	*ALL, *HELD, *NOTSTARTED, *SCHEDULED, *WAIT, *READY, *STARTED, *SENT, *NOTSENT, *RUNNING, *ENDING, *ENDED	Optional
ENDCODE	End code	0-99, *ANY, *FAIL, *SUCCESS, *NOTRUN	Optional

Top

---

## Change request description (CRQ)

Specifies the change request name and sequence number of the activity list to be worked with.

The possible change request name value is:

*change-request-name*

Specify the name of the change request for which activities are to be displayed.

The possible sequence number value is:

*sequence-number*

Specify the sequence number of the change request.

Top

---

## Status (STATUS)

Specifies the status for which to display change request activities.

The possible values are:

**\*ALL** All change request activities are displayed regardless of status.

**\*HELD**  
Displays the status when the user holds the node.

**\*NOTSTARTED**  
The node was not processed by the application error. You can find further information about the error in the Change Request Log.

**\*SCHEDULED**  
The activity for the node begins at the time you have specified. Sometimes you may receive this status when one or more conditions still must be specified.

**\*WAIT**  
The node is waiting for you to specify one or more conditions. Until you specify the needed conditions, the activity does not run.

**\*READY**  
The node is ready to start. You have defined all necessary conditions and schedules.

**\*STARTED**  
Processing of an activity has started at the central site system.

**\*RUNNING**  
The node currently is running on the local system. SNADS is attempting to deliver the entry to the corresponding system. If the activity remains in this status for a long time, you may have a configuration error or SNADS problem. First, check  
  
First, check the distribution queue at the remote site. Second, make sure that all QMSF jobs are running under the QSYSWRK subsystem. Third, make sure that SUBSYSTEM QSNADS is running.

**\*ENDING**  
The system displays this status when the user tries to end the activity for the node that was previously in STARTED or RUNNING status. The end is in progress.

**\*ENDED**  
The note that was running has ended.

**\*SENT**  
Displays only the activities with a status of Sent.

**\*NOTSENT**  
Displays only the activities with a status of Notsent.

Top

---

## End code (ENDCODE)

Specifies the end code for which to display the change request activities.

The possible values are:

**\*ANY** Display all activities regardless of end codes.

**\*FAIL** Displays all activities with the highest end code between 10 and 89.

**\*SUCCESS**

Displays all activities with the highest end code between 0 and 9.

**\*NOTRUN**

Displays all activities with the highest end code between 90 and 99.

***end-code***

Specify the end code to use. Displays activities with an end code greater than or equal to the one specified. The valid values range from 0 through 99.

[Top](#)

---

## **Examples**

WRKSBMCRQA CRQ(CHG001 123456)

This command works with the activities for change request CHG001 with sequence number 123456.

[Top](#)

---

## **Error messages**

None

[Top](#)



---

## Work with Supported Products (WRKSPTPRD)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Work with Supported Products (WRKSPTPRD) command provides a list of products for you. You can specify products, options, and language features to process by way of service requests and program temporary fix (PTF) orders.

Top

---

### Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *_PRINT	Optional, Positional 1
RLS	Release level	Character value, *CURRENT, *ALL	Optional

Top

---

### Output (OUTPUT)

Specifies whether the output from the command is shown at the requesting workstation or printed with the job's spooled output.

\*     The output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

**\*PRINT**  
The output is printed with the job's spooled output.

Top

---

### Release level (RLS)

Specifies the release level of products to work with. Multiple releases of products can be supported each release must be specified on a separate WRKSPTPRD command.

**\*CURRENT**  
Support information is displayed for the releases of those products known to be compatible with the currently installed release of the operating system. More than one release of a single licensed program might be displayed.

#### Note:

Programming Request for Price Quotations (PRPQs), Licensed Program Offerings (LPOs) and System Manager packaged products cannot be displayed or printed using \*CURRENT. Specify the release level to display or print support information for these products.

**\*ALL** Support information for all releases of products is printed. If OUTPUT(\*PRINT) is specified,

support information for all releases of products is printed. If OUTPUT(\*) is specified, support information for all releases of products is displayed

#### *release-level*

Specify the release level in VxRxMx format where V represents version, R represents release, and M represents modification.

Top

---

## Examples

### Example 1: Showing Menu Panel for Current Release Level

```
WRKSPTPRD
```

This command displays the Work with Supported Products panel for the current release level of each product.

### Example 2: Showing a Menu for Products at a Specific Release Level

```
WRKSPTPRD  RLS(V5R2M0)
```

This command displays the Work with Supported Products panel for products at the release level of Version 5, Release 2, Modification 0.

### Example 3: Printing a List for All Supported Releases

```
WRKSPTPRD  OUTPUT(*PRINT)  RLS(*ALL)
```

This command prints a list of supported products and product options for all supported releases.

Top

---

## Error messages

### \*ESCAPE Messages

SMU15A5

Unexpected error processing WRKSPTPRD command.

Top



---

## Work with Service Requesters (WRKSRVRQS)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

[Parameters](#)  
[Examples](#)  
[Error messages](#)

The Work with Service Requesters (WRKSRVRQS) command defines a list of service requesters entitled to receive service from this system.

### Restriction:

This command is shipped with public \*EXCLUDE authority and the QSRV and QSRVBAS user profiles have private authorities to use the command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

WRKSRVRQS

This command displays the Work with Service Requesters panel from which you can add, change, copy, display, or remove information about the service requesters that you support.

[Top](#)

---

### Error messages

None

[Top](#)



---

## 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  
500 Columbus Avenue  
Thornwood, NY8809  
U.S.A.

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 be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department 49XA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Printing  
AFP  
AS/400  
CICS  
COBOL/400  
C/400  
DataPropagator  
DB2  
IBM  
Infoprint  
InfoWindow  
iSeries  
LPDA  
OfficeVision

OS/400  
Print Services Facility  
RPG/400  
SystemView  
System/36  
TCS  
WebSphere

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

---

## Terms and conditions for downloading and printing publications

Permissions for the use of the publications you have selected for download are granted subject to the following terms and conditions and your indication of acceptance thereof.

**Personal Use:** You may reproduce these Publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these Publications, or any portion thereof, without the express consent of IBM.

**Commercial Use:** You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations. IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

All material copyrighted by IBM Corporation.

By downloading or printing a publication from this site, you have indicated your agreement with these terms and conditions.

---

## Code disclaimer information

This document contains programming examples.

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

All sample code is provided by IBM for illustrative purposes only. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

All programs contained herein are provided to you "AS IS" without any warranties of any kind. The implied warranties of non-infringement, merchantability and fitness for a particular purpose are expressly disclaimed.





Printed in USA