



@server

iSeries

CL Commands Volume 6





@server

iSeries

CL Commands Volume 6

Contents

Command Descriptions	1
CHGTCPHTE (Change TCP/IP Host Table Entry) Command Description	1
CHGTCPIFC (Change TCP/IP Interface) Command Description	4
CHGTCPRTE (Change TCP/IP Route) Command Description	12
CHGTFTPA (Change TFTP Server Attributes) Command Description	16
CHGUSFDEVE (Change Ultimedia System Facilities Device Entry) Command Description	19
CHGUSRAUD (Change User Audit) Command Description	22
CHGUSRPTI (Change User Print Information) Command Description	24
CHGUSRPRF (Change User Profile) Command Description	25
CHGUSRTRC (Change User Trace Buffer) Command Description	43
CHGVAR (Change Variable) Command Description	44
CHGWSE (Change Work Station Entry) Command Description	49
CHGWTR (Change Writer) Command Description	52
CHKASPBAL (Check ASP Balance) Command Description	56
CHKCMNTRC (Check Communications Trace) Command Description	56
CHKIGCTBL (Check DBCS Font Table) Command Description	57
CHKDKT (Check Diskette) Command Description	58
CHKDLO (Check Document Library Object) Command Description	61
CHKEXPBRM (Check Expired Media for BRM) Command Description	63
CHKIN (Check In) Command Description	64
CHKOBJ (Check Object) Command Description	65
CHKOBJITG (Check Object Integrity) Command Description	69
CHKOUT (Check Out) Command Description	73
CHKPWD (Check Password) Command Description	75
CHKPRDOPT (Check Product Option) Command Description	75
CHKRCDLCK (Check Record Locks) Command Description	77
CHKTAP (Check Tape) Command Description	78
CLRDKT (Clear Diskette) Command Description	82
CLRJOBQ (Clear Job Queue) Command Description	83
CLRLIB (Clear Library) Command Description	84
CLRMSGQ (Clear Message Queue) Command Description	86
CLROUTQ (Clear Output Queue) Command Description	87
CLRPFM (Clear Physical File Member) Command Description	88
CLRPOOL (Clear Pool) Command Description	90
CLRSVAVF (Clear Save File) Command Description	91
CLRSVRSEC (Clear Server Security Data) Command Description	92
CLRTRCDTA (Clear Trace Data) Command Description	93
CLOF (Close File) Command Description	93
COMMIT (Commit) Command Description	94
CMPJRNIMG (Compare Journal Images) Command Description	96
CPROBJ (Compress Object) Command Description	102
CFGDEVMLB (Configure Device Media Library) Command Description	106
CFGDSTSRV (Configure Distribution Services) Command Description	110
CFGHTTPSCH (Configure HTTP Search) Command Description	112
CFGPM400 (Configure PM/400) Command Description	118
CFGTCPPTP (Configure Point-to-Point TCP/IP) Command Description	118
CFGSYSSEC (Configure System Security) Command Description	119
FGTCP (Configure TCP/IP) Command Description	119
FGTCPAPP (Configure TCP/IP Applications) Command Description	120
FGTCPBP (Configure TCP/IP BOOTP) Command Description	121
FGTCPHTTP (Configure TCP/IP HTTP) Command Description	122
FGTCPRTD (Configure TCP/IP RouteD) Command Description	122
FGTCPNMP (Configure TCP/IP SNMP) Command Description	122

CVTCLSRC (Convert CL Source) Command Description	123
CVTDAT (Convert Date) Command Description	125
CVTDIR (Convert Directory) Command Description	129
CVTDLSNAM (Convert Document Library Services Name) Command Description	132
CVTEDU (Convert Education) Command Description	135
CVTIPSIFC (Convert IP over SNA Interface) Command Description	136
CVTIPSLOC (Convert IP over SNA Location Entry) Command Description	137
CVTOPTBKU (Convert Optical Backup) Command Description	138
CVTPFRDTA (Convert Performance Data) Command Description	140
CVTPFRTHD (Convert Performance Thread Data) Command Description	142
CVTRPCSRC (Convert RPC Source) Command Description	144
RPCGEN (Convert RPC Source) Command	145
CVTTCPCL (Convert TCP/IP CL Source) Command Description	146
COPY (Copy) Command Description	148
CPY (Copy) Command Description	148
CPYCFGL (Copy Configuration List) Command Description	156
CPYIGCTBL (Copy DBCS Font Table) Command Description	157
CPYDSTRPSO (Copy Distribution Repository Object) Command Description	163
CPYDOC (Copy Document) Command Description	170
CPYF (Copy File) Command Description	172
CPYFRMDIR (Copy From Directory) Command Description	194
CPYFRMDKT (Copy From Diskette) Command Description	195
CPYFRMIMPF (Copy From Import File) Command Description	200

Command Descriptions

CHGTCPHTE (Change TCP/IP Host Table Entry) Command Description

CHGTCPHTE Command syntax diagram

Purpose

The Change TCP/IP Host Table Entry (CHGTCPHTE) command is used to change the host names and text description fields for an existing host table entry in the local host table. A host table entry consists of one internet address, up to four host names, and one text description.

The CHGTCPHTE command can change a minimum of zero and a maximum of four host names associated with a specific internet address. This command can also be used to add or remove a host-name value associated with a specific internet address. To remove a host-name value, specify *BLANK as the host name. Setting all the host names for a host table entry to *BLANK is not allowed.

If the CHGTCPHTE command is prompted with an internet address specified, the current host names and text description for the host table entry associated with that internet address are displayed in the appropriate prompt fields.

If a remote name server is being used by your iSeries 400, the search order used (whether the remote name server or local host table is searched first) for a resolution between a host name and an internet address depends on how the searched-first value was configured on the configuration panel of the remote name server. To change the search order, use option 13 on the Configure TCP/IP (CFGTCP) command.

The TCP/IP host table is shipped with the loopback entry. This entry has an internet address of 127.0.0.1 and two host names: LOOPBACK and LOCALHOST. The loopback host name can be associated only with an internet address that has a first-byte value equal to 127.

Related APPC over TCP/IP Information:

APPC over TCP/IP (part of the AnyNet/400* function) uses the host name to map location names to internet addresses. The host name must be in the form:

```
location.netid.SNA.IBM.COM
```

Where *location* is the remote location the program is opening to, and *netid* is the network identifier for this connection. *SNA.IBM.COM* is the qualifier that designates this as the APPC over TCP/IP domain.

Location names support characters that cannot be present in host names (for example: \$ (dollar), @ (at sign), and # (number sign)). Therefore, the APPC application can open only to locations that fulfill the TCP/IP host name syntax. This limits location names used for APPC over TCP/IP to the characters A-Z (uppercase and lowercase) and 0-9.

Restriction: You must have *IOSYSCFG special authority to use this command.

Required Parameter

INTNETADR

Specifies the internet address associated with the host name (or names) or the text-description field that is to be changed in the local host table. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Optional Parameters

HOSTNAME

Specifies the host names corresponding to the internet address. The host name can be either the short form or the full domain version of the name. A common practice is to define one short name that is unique within your local network and to also define the full domain version of the host name that is unique within the internet. Specify from 1 to 4 different host names to be associated with the internet address. Host names may be up to 255 characters in length.

A domain name or a host name can be a text string having 1 to 255 characters. Domain names consist of one or more labels separated by periods. Each label can contain up to 63 characters. The first character of each label must be an alphabetic character or a digit. The last character of each label must be an alphabetic character, a digit, or a period. The following characters are allowed in domain names:

- Alphabetical characters A through Z
- Digits 0 through 9
- Underscore (_)
- Minus sign (-)
- Period (.). Periods are allowed only when they separate labels of the domain style name or as the last character in the domain name. (Refer to RFC 1034.) A domain name cannot have two consecutive periods.

Note:

These characters are part of the Syntactic Character Set (character set number 640). This character set is also commonly referred to as invariant.

Other domain name and host name conventions include the following:

- Uppercase and lowercase characters are allowed, but no significance is attached to the case. The host name (HOSTNAME) may be converted to uppercase depending on the combination of characters and digits. If the HOSTNAME is surrounded with apostrophes ('), the case is maintained as entered.
- The host name returned when searching the host table for an internet address is the first host name associated with the internet address. For example, if the address 9.130.38.187 is defined in the host table with names ROCHESTER, JOHN, and RCHAS100, the name ROCHESTER would be returned. The other two host names would not be used in this type of search. However, these host names would be used when searching the host table to find the internet address associated with the names JOHN and RCHAS100.
- Try to limit your domain name labels to 12 characters. Shorter labels are easier to remember.
- It is a common practice to use hierarchical names that allow predictable extensions for change and growth. Domain names normally reflect the delegation of authority or hierarchy used to assign them.

For example, the name SYS1.MFG.ABC.COM can be broken down into the following:

COM All commercial networks.

ABC.COM

All systems in the ABC company's commercial network.

MFG.ABC.COM

All manufacturing systems in the ABC company's commercial network.

SYS1.MFG.ABC.COM

A host named SYS1 in the manufacturing area of the company's commercial network.

The COM designation is one of several domain names used by convention when connecting to the Internet. Some of the other domain names that follow this convention are:

COM Commercial organizations
EDU Educational institutions
GOV Government institutions
MIL Military groups
NET Major network support centers
ORG Organizations other than those listed previously
ARPA Temporary ARPANET domain
Country or region code
 Countries other than USA

***SAME:** This host-name value is not to be modified.

Note:

If *SAME is specified and no other host-name values are specified, all of the host-name values remain the same. If a host table entry has more than one host name identified and if the first host name is specified but no other element values are specified, the remaining host names are not changed.

***BLANK:** This host-name value is changed to blanks if it previously existed.

host-name: Specify a host name to be associated with the specified internet address that replaces the current host-name value. When running APPC over TCP/IP, *name* is in the form:

location.netid.SNA.IBM.COM

TEXT Specifies a comment associated with this host table entry.

Note:

If the host table will be copied to a system using a different code page than the system it was created on, it is suggested that you avoid using certain characters in a comment. Host table entry comments will be more portable if they are limited to characters in the Syntactic Character Set (invariant).

***SAME:** The text-description field for this host table entry is not to be modified.

***BLANK:** The text-description field for this host table entry is to be changed to blanks.

'description': Specify a text-description field to be associated with the specified internet address. Comments can contain a maximum of 64 characters.

Examples for CHGTCPHTE

Example 1: Changing a Host Name

```
CHGTCPHTE INTNETADR('132.28.71.5')
HOSTNAME((*SAME) (*SAME) (NEWAS400HOST))
TEXT(*BLANK)
```

This command changes the third host name associated with internet address 132.28.71.5 to NEWAS400HOST but does not modify the first, second, or fourth host names. The text of the descriptive comment for this host table entry is set to blanks.

Example 2: Changing All Host Names

```
CHGTCPHTE INTNETADR('9.130.25.21')
  HOSTNAME((MYHOST) (MYHOST.MYNET)
           (MYHOST.MYNET.MYCORP)
           (MYHOST.MYNET.MYCORP.MYFIELD))
  TEXT(*SAME)
```

This command changes all host names associated with internet address 121.14.32.5. The first host name is specified in the short form, MYHOST. The fourth host name is specified in the fully qualified form, MYHOST.MYNET.MYCORP.MYFIELD. The descriptive comment for this host table entry is not changed.

Example 3: Changing Host Names and Text Description

```
CHGTCPHTE INTNETADR('132.28.71.5')
  HOSTNAME((AS400ETH.SALES.ABC.COM) (AS400ETH.SALES.ABC)
           (*BLANK) (*BLANK))
  TEXT('THIS ENTRY UPDATED ON 19 FEB 1994 BY T.J.')
```

This command changes the first and second host names associated with internet address 132.28.71.5 to AS400ETH.SALES.ABC.COM and AS400ETH.SALES.ABC. The third and fourth host names, if they existed, are changed to blanks. The descriptive comment for this host table entry is changed to 'THIS ENTRY UPDATED ON 19 FEB 1994 BY T.J.'.

Error messages for CHGTCPHTE

*ESCAPE Messages

TCP1901

Internet address &1 not valid.

TCP1902

Internet address &1 not valid.

TCP1903

Specified host name not valid.

TCP1907

Internet address entry &1 does not exist.

TCP1908

Internet address &1 not valid.

TCP1910

LOOPBACK internet address &1 not valid.

TCP1929

Host table not available.

TCP1936

All host names for internet address &1 are blank.

CHGTCPIFC (Change TCP/IP Interface) Command Description

CHGTCPIFC Command syntax diagram

Purpose

The Change TCP/IP Interface (CHGTCPIFC) command is used to change an existing interface in the Transmission Control Protocol/Internet Protocol (TCP/IP) configuration. The interfaces defined by the CHGTCPIFC command are logical interfaces. They are not physical interfaces. Each interface is associated with a line description. The line description is the physical connection from the iSeries 400 to the TCP/IP network.

The iSeries 400 TCP/IP implementation supports *multihoming*. This allows you to specify either a single interface or multiple interfaces per line description. You can have your iSeries 400 appear as any one or combination of the following:

- A single host on a network over a communications line
- Multiple hosts on the same network over the same communications line
- Multiple hosts on different networks over the same communications line
- Multiple hosts on the same network over multiple communications lines
- Multiple hosts on different networks over multiple communications lines

Notes:

1. If you attempt to change a value for an interface that will invalidate a route or remote system information (RSI) associated with the interface, the change will not be allowed.
2. In SNMP, an interface is a physical interface. The physical interface relates directly to an input/output processor (IOP).
3. The interface table is shipped with a default interface of 127.0.0.1. The line description value associated with the 127.0.0.1 interface is *LOOPBACK. The host table is also shipped with an entry that has an internet address of 127.0.0.1 and host names of LOOPBACK and LOCALHOST.

Attention:

Before attempting to start an X.25 interface, ensure that the remote system information (RSI) for non-DDN X.25 interfaces that use a permanent virtual circuit (PVC) is configured. Use the Add TCP/IP Remote System Information (ADDTCPRSI) command to do this. Incoming data from a remote system on the X.25 network is not processed unless an RSI entry for the PVC is configured on the X.25 interface before the interface is started.

Restrictions:

- You must have *IOSYSCFG special authority to use this command.
- Only certain values can be changed using this command. The values that can be changed depend on the status of the interface, the status of the dependent routes, and the remote system information configured.

Required Parameter

INTNETADR

Specifies the internet address. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Optional Parameters

LIND Specifies the name of the line description associated with the to be changed interface. The following conditions are based on the interface type that the user defines:

Token-ring

The name must be previously defined on the Create Line Description (Token-Ring Network) (CRTLINTRN) command.

X.25 The name must be previously defined on the Create Line Description (X.25) (CRTLINX25) command.

Ethernet

The name must be previously defined on the Create Line Description (Ethernet) (CRTLINETH) command.

DDI The name must be previously defined on the Create Line Description (DDI Network) (CRTLINDDI) command.

Frame relay

The name must be previously defined on the Create Line Description (Frame Relay Network) (CRTLINFR) command.

Wireless

The name must be previously defined on the Create Line Description (Wireless Network) (CRTLINWLS) command.

Twinax (TDLC)

The name must be previously defined on the Create Line Description (CRTLINTDLC) command.

TCP/IP can also be used on certain line descriptions attached to these network interfaces (NWI):

- An ISDN NWI using an X.25 line description.
 - The ISDN NWI is created using the Create Network Interface ISDN (CRTNWIISDN) command.
 - The X.25 line is created using the Create Line X.25 (CRTLINX25) command and attached to the ISDN NWI by specifying the NWI, NWICHLTYPE, NWICHLNBR, and SWTNWILST parameters.
- A frame relay NWI using a frame relay, token ring, Ethernet, or DDI line description.
 - The frame relay NWI is created using the Create Network Interface Frame Relay Network (CRTNWIFR) command.
 - The line description is created using the appropriate Create Line command and attached to the frame relay NWI by specifying the NWI and NWIDLCL parameters.

***SAME:** The same line description existing for this interface is used.

***LOOPBACK:** The interface being changed by this command is the loopback or LOCALHOST interface. Because processing associated with loopback does not extend to a physical line, there is no line description associated with a loopback address. This special value must be used for any internet address that has a first octet value of 127.

***VIRTUALIP:** This special value is used if you are adding a 'circuitless' interface. This means that this interface has a real IP address but is NOT tied to any physical hardware. Interfaces of this type are useful to identify the address of the iSeries 400 to remote systems over point-to-point links (PPP, SLIP, Frame Relay, X.25).

***OPC:** This special value is used if you are adding an OptiConnect interface over TCP/IP. This interface is attached to the optical bus (OptiConnect).

line-description: Specify the line description to be used for this interface.

SUBNETMASK

Specifies the subnet mask, which is a bit mask that defines the part of the network where this interface attaches. The mask is a 32-bit combination that is logically ANDed with the internet address to determine a particular subnetwork. The bits of the mask set to the value one (1) determine the network and subnetwork portions of the address. The bits set to the value zero (0) determine the host portion of the address.

Note:

The network portion must be equal to one bits in the subnetmask. The host portion of an address must be at least two bits wide.

The bits that identify the subnetwork are not required to be adjacent in the address. However, if this subnet mask value is changed, it might invalidate or affect the routes using this interface. To prevent this, keep the subnet bits contiguous and located in the most significant bits of the host address.

***SAME:** The same subnetmask existing for this interface is used.

***HOST:** The subnetmask value used will be 255.255.255.255. *HOST is not valid for X.25 interfaces.

subnet-mask: Specify the mask for the network field and host address field of the internet address that defines a subnetwork. The subnet mask is in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. The subnet mask must mask off all bits of the network class's network ID portion of the internet address.

For more detailed information on subnet masks and an example, see the Add TCP/IP Interface (ADDTCPIFC) command description.

LCLIFC

The local IP interface that the internet address defined in INTNETADR will be associated with.

The associated local interface is used to allow for transparent subnetting or unnumbered networks on the iSeries 400. Any local interface may be used for LCLIFC, except for interfaces defined for the X.25 or PPP linetypes.

***SAME:** The current associated local interface is used.

***NONE:** No associated local interface is used.

local-interface: Specify an associated local interface for the interface being changed. Note that the specified associated local interface must already exist.

TOS

Specifies the type of service to be used. The type of service defines how the internet hosts and routers should make trade-offs between throughput, delay, reliability, and cost.

***SAME:** The type of service does not change.

***NORMAL:** Normal service is used for delivery of data.

***MINDELAY:** Minimize delay means that prompt delivery is important for data on this connection.

***MAXTHRPUT:** Maximize throughput means that a high data rate is important for data on this connection.

***MAXRLB:** Maximize reliability means that a higher level of effort to ensure delivery is important for data on this connection.

***MINCOST:** Minimize monetary cost means that lower cost is important for data on this connection.

MTU Specifies the maximum size (in bytes) of IP datagrams that can be transmitted through this interface. A datagram is a basic unit of information passed over an internet network. The minimum size of any maximum transmission unit value is 576 bytes. If this value is changed it affects the MTUs of routes using this interface.

***SAME:** The existing maximum transmission unit value for this interface is used.

***LIND:** The MTU is determined by the information specified in the line description. If *LIND is specified, the MTU is equal to the largest amount of data that can be transmitted on the line. If the LIND parameter specifies *LOOPBACK, *VIRTUALIP, or *OPC, then the MTU value will be:

***LOOPBACK**

576

***VIRTUALIP**

576

***OPC** 32768

maximum-transmission-unit: Specify a value for the maximum transmission unit in bytes. The maximum MTU that can be specified for this interface depends on the type of physical connection to the network. The following table lists the maximum MTU values that can be specified based on the line type:

X.25 4096

Token ring (4 meg)

4060

Token ring (16 meg)

16388

Ethernet 802.3

1492

Ethernet Version 2

1500

DDI 4352

Frame relay

8177

Wireless 802.3

1492

Wireless Version 2

1500

Twinax (TDLC)

4105

Notes:

1. It is suggested that the same MTU values be used for all interfaces on the same network.
2. The actual MTU value used for an interface is resolved during interface activation. This value is the minimum of either the specified MTU value for the interface or the largest amount of data that can be transmitted on the line.
3. The same MTU value does not need to be specified for all interfaces defined on the same subnet. However, all interfaces must have an MTU that does not exceed the value used when *LIND is specified for the interface MTU.
4. To view the MTU value actually used for an interface, do the following:
 - a. Use the ADDTCPIFC command to add the interface.

- b. Use the Start TCP/IP Interface (STRTCPIFC) command to activate the interface.
- c. Use the Work with TCP/IP Status (WRKTCIPSTS or NETSTAT) command to view the actual MTU value of the interface in bytes.

AUTOSTART

Specifies whether the interface is automatically started when the TCP/IP stack is activated by using the Start TCP/IP (STRTCP) command.

***SAME:** The existing autostart value for this interface is used.

***YES:** The interface is automatically started at STRTCP time.

***NO:** The interface is not started at STRTCP time.

Note:

The Start TCP/IP Interface (STRTCPIFC) command can be used to start an interface any time after TCP/IP has been activated.

PVCLGLCHLI

Specifies the permanent virtual circuit (PVC) logical channel identifiers that can be established on an X.25 interface by the TCP/IP protocol stack. Up to 64 unique channel identifiers may be specified. These logical channel identifiers must exist in the X.25 line description.

With this parameter you can share the line with other communications software, such as Systems Network Architecture (SNA). It prevents the TCP/IP protocol stack from monopolizing the PVCs defined for the line.

Notes:

1. This parameter is valid only for an interface defined on an X.25 line description.
2. PVCs cannot be used in a DDN network.
3. When specifying PVCs for an X.25 interface, all interfaces on the same X.25 network must have this same set of PVC logical channel identifiers specified. This is especially important if one or more remote system information (RSI) entries will use a PVC to connect to the RSI entry's remote system on the X.25 network.
4. If the RSI entries are defined such that two or more remote internet addresses can be reached across the same PVC, that PVC is shared.
5. The sum of the maximum switched virtual circuits (MAXSVC) and the number of PVCs cannot exceed 64.

***SAME:** The existing PVC logical channel identifier values for this interface are used.

***NONE:** All existing PVC logical channel identifier values for this interface are removed. If no PVC values are defined, *NONE is shown.

logical-channel-identifier: Specify the PVC logical channel identifier value. The value may be from 001 to FFF. Up to 64 PVC logical channel identifiers can be specified.

IDLVCTTIMO

Specifies the duration (in seconds) that the TCP/IP Network Access Manager (NAM) waits before clearing an idle virtual circuit established on an X.25 link. Clearing an idle virtual circuit frees resources on the network. TCP/IP automatically reestablishes virtual circuits when required to send or receive data. Virtual circuits are transparent to a TCP/IP client and have no noticeable effect on TCP connections.

Note: This parameter is valid only for switched virtual circuits (SVCs) on an interface defined on an X.25 line description. It is not valid for permanent virtual circuits (PVCs).

***SAME:** The existing idle virtual circuit timeout value for this interface is used.

number-of-seconds: Specify the idle virtual circuit timeout. Valid values range from 1 through 600 seconds.

MAXSVC

Specifies the maximum number of concurrent switched virtual circuits (SVC) that can be established on an X.25 interface by the TCP/IP protocol stack.

With this parameter you can share the line with other communications software, such as Systems Network Architecture (SNA). It prevents the TCP/IP protocol stack from monopolizing the SVCs defined for the line. This parameter is valid only for an interface defined on an X.25 line description.

Note: The sum of the maximum switched virtual circuits (MAXSVC) and the number of PVCs cannot exceed 64.

***SAME:** The existing maximum SVC value for this interface is used.

X.25-maximum-virtual-circuits: Specify the number of SVCs that the TCP/IP protocol stack can use simultaneously. The valid values range from 0 through 64. If 64 is specified, the number of SVCs that are configured is calculated by adding the number of *SVCIN, *SVCOUT and *SVCBOTH SVCs defined for the line description (LIND) being used by this interface. This is the maximum number of SVCs that can be authorized for processing by the TCP/IP protocol stack.

DDN Specifies whether the X.25 interface is connected to the Defense Data Network. The DDN network is a special type of X.25 network used by TCP/IP customers with special security needs.

Note: This parameter is valid only for switched virtual circuits (SVCs) on an interface defined on an X.25 line description. It is not valid for permanent virtual circuits (PVCs).

Attention:

If you specify multiple interfaces to the same X.25 network, the DDN value should be equal for all of those interfaces. This is not enforced by the ADDTCPIFC or CHGTCPIFC commands.

If the X.25 network is the DDN network, do not define the remote system information for any of the remote systems on the network. The remote system information for the DDN X.25 network is determined from the destination IP address.

***SAME:** The existing DDN value for this interface is used.

***NO:** The X.25 interface is not connected to the Defense Data Network.

***YES:** The X.25 interface is connected to the Defense Data Network.

BITSEQ

Specifies the order, most or least significant bit first, in which the Address Resolution Protocol (ARP) places the bits in the hardware address. This parameter is valid only for a token-ring local area network (TRLAN) line.

Note:

All interfaces defined to a single token-ring line must have the same BITSEQ value. This is checked by the CHGTCPIFC code to ensure consistent values.

***SAME:** The existing bit sequence value for this interface is used.

***MSB:** The most significant bit is placed first.

***LSB:** The least significant bit is placed first.

Examples for CHGTCPIFC

Example 1: Changing Autostart Value

```
CHGTCPIFC INTNETADR('130.14.3.5')
AUTOSTART(*NO)
```

This command assumes that an interface identified by 130.14.3.5 exists. This command changes the autostart value from *YES to *NO. The interface is not automatically started when the STRTCP command is entered.

Example 2: Changing MAXSVC and IDLVCTTIMO

```
CHGTCPIFC INTNETADR('8.77.0.21')
INDLVCTTIMO(45) MAXSVC(15)
```

This command changes the idle virtual circuit time-out to 45 seconds and the maximum number of concurrent SVCs allowed to be used by TCP/IP on this interface to 15.

Example 3: Change an interface for a twinax line that is using an associated local interface

```
CHGTCPIFC INTNETADR('199.1.1.99')
LCLIFC('199.1.1.1')
```

This command will change the TCP/IP interface for the twinax line named TDLCLINE. This interface will be associated with local interface 199.1.1.1. This means that the devices attached to twinax line 199.1.1.99 can take advantage of 'appearing' to be on the same network as the local 199.1.1.1 interface (transparent subnetting). No special routing is required to ensure packets from the twinax connected hosts can travel to the local 199.1.1.0 network. Also, hosts on the 199.1.1.0 network can also reach the twinax hosts without any additional routing on the host systems.

Error messages for CHGTCPIFC

*ESCAPE Messages

TCP1D03

&1 member record length not correct.

TCP1D04

Error occurred processing member &1 of &2/&3.

TCP1901

Internet address &1 not valid.

TCP1902

Internet address &1 not valid.

TCP1908

Internet address &1 not valid.

TCP8050

*IOSYSCFG authority required to use &1.

TCP9999

Internal system error in program &1.

CHGTCPRTE (Change TCP/IP Route) Command Description

CHGTCPRTE Command syntax diagram

Purpose

The Change TCP/IP Route (CHGTCPRTE) command is used to change an existing route in the Transmission Control Protocol/Internet Protocol (TCP/IP) configuration.

Five parameter values uniquely define a route. These values are the route destination (RTEDEST), the subnet mask (SUBNETMASK), the type of service (TOS), the internet address of the next system on the route (NEXTHOP), and the preferred binding interface (BINDIFC). For default routes and default multicast routes (*DFTRROUTE and *DFTMCAST), the NEXTHOP, TOS, and BINDIFC values uniquely define the route because the SUBNETMASK is always *NONE.

Restrictions:

1. You must have *IOSYSCFG special authority to use this command.
2. Only one parameter, the MTU value, can be changed on an existing route entry. The route cannot be in use when attempting to change its MTU value.
3. Attempts to change a route that is required to reach an existing RSI entry will fail.

Required Parameters

RTEDEST

Specifies the route destination being changed. You must specify all 4 bytes that make up an internet address though some of the bytes may be equal to 0. For example, a route to all the hosts on the 9.5.11 subnetwork is identified by entering 9.5.11.0 for the route destination. Used in combination with a subnetmask, type of service value, and next hop, the route destination uniquely identifies a route to a network or system.

***DFTRROUTE:** Specifies that a default route entry is being changed. A default route entry is used by the system to route data that is being sent to a remote destination that does not have a specific route defined. The system allows a maximum of 8 default route entries. The default route entries are used based on the availability of the next hop gateway and the type of service (TOS). If the application requests a specific TOS, the TOS of the default route used must match the TOS requested. If no default route is found that matches the requested TOS, the first available default route with a TOS of *NORMAL is used.

***DFTMCAST:** Specifies that a default multicast route entry is being changed. A default multicast route entry is used by the system to select a local interface when sending data to a multicast group. The default multicast entry is used when the application does not specifically name the local interface over which multicast packets should be sent. When RTEDEST(*DFTMCAST) is specified, SUBNETMASK(*NONE) must be specified and the NEXTHOP parameter must be the internet address of an interface that had previously been added with the Add TCP/IP Interface (ADDTCPIFC) command.

route-destination: Specify the route destination being changed. The route destination can be specified in the form *nnn.0.0.0*, for Class A, *nnn.nnn.0.0* for Class B, and *nnn.nnn.nnn.0* for Class C, or *nnn.nnn.nnn.nnn* for any combination thereof, where *nnn* is a decimal number ranging from 0 through 255.

Any combination thereof means that you may specify a route, such as 9.5.0.0 to the hosts on the 9.5 subnet, even though all 9.5.x.x addresses are class A network addresses.

Exceptions:

- The first byte (octet) must be greater than 0 and less than 255.
- The last byte (octet) may not equal 255.
- The last byte (octet) may not equal 0 if *HOST is specified for the SUBNETMASK value.
- Routes to a broadcast address are not allowed.

SUBNETMASK

Specifies a bit mask that identifies to TCP/IP which bits of the value specified for the route destination (RTEDEST) compose the network and subnet portions of the internet address. By defining the network portion and subnetwork portion of the RTEDEST address, the subnet mask also defines which bits of the RTEDEST address make up the host portion. The mask is a 32-bit combination that is logically ANDed with the internet address to determine a particular subnetwork. The bits of the mask set to the value one (1) determine the network and subnetwork portions of the address. The bits set to the value zero (0) determine the host portion of the address.

***NONE**: There is no subnet mask. If RTEDEST(*DFTRROUTE) or RTEDEST(*DFTMCAST) is specified, SUBNETMASK(*NONE) must be specified. *NONE is valid only for the *DFTRROUTE and *DFTMCAST route destination values.

***HOST**: The internet address value specified in the route destination field is a host address. The subnetmask value is calculated to be 255.255.255.255.

subnet-mask: Specify the mask of the subnet field. The internet address is in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. For example, a destination route's internet address value of 129.35.11.0 is a Class B subnet. The network ID part of its address is 129.35. The upper 2 bytes must designate 255 in the subnetmask. The subnetmask must appear like 255.255.x.x, where x is determined by the user. The portion of the subnetmask that is associated with the network portion of a particular class of address must equal 255.

TOS Specifies the type of service to be used. The type of service defines how the internet hosts and routers should make trade-offs between throughput, delay, reliability, and cost.

***NORMAL**: Normal service is used for delivery of data.

***MINDELAY**: Minimize delay means that prompt delivery is important for data on this connection.

***MAXTHRPUT**: Maximize throughput means that a high data rate is important for data on this connection.

***MAXRLB**: Maximize reliability means that a higher level of effort to ensure delivery is important for data on this connection.

***MINCOST**: Minimize monetary cost means that lower cost is important for data on this connection.

NEXTHOP

Specifies the internet address of the next system (gateway) on the route.

internet-address: Specify the internet address of the next system on the route. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Optional Parameters

MTU Specifies the maximum size (in bytes) of IP datagrams that can be transmitted through this route. A datagram is a basic unit of information passed over an internet network. The minimum size of any maximum transmission unit value is 576 bytes.

***SAME:** The existing maximum transmission unit value for this route is used.

***IFC:** The maximum transmission unit (MTU) is the MTU of the interface that is associated with this route.

maximum-transmission-unit: Specify a value for the maximum transmission unit in bytes. The maximum MTU that can be specified for this route depends on the type of physical connection to the network. The following table lists the maximum MTU values that can be specified based on the line type:

X.25 4096

Token ring (4 meg)
4060

Token ring (16 meg)
16388

Ethernet 802.3
1492

Ethernet Version 2
1500

DDI 4352

Frame relay
8177

Wireless 802.3
1492

Wireless Version 2
1500

Twinax (TDLC)
4105

Notes:

1. TCP/IP uses the route MTU value to calculate the size of the datagrams it sends. For best performance, specify a value that is no smaller than the smallest MTU used by host systems along the entire path of this route. If this information is not available, use the default value of 576.
2. The MTU of a route cannot exceed the MTU of the interface on which the NEXTHOP value is accessed. If the interface's MTU value was specified as *LIND, the interface's MTU value is derived from the line description. If the route's MTU value is specified as *IFC and the interface's MTU value is specified as *LIND, both values are derived from the line description.
3. The actual MTU value used for a route is resolved during interface activation. This value is the minimum of either the specified MTU value for the route or the MTU value determined from the associated interface used by the route.

METRIC

Specifies the 'cost' associated with the use of this route. A value of 1 is a route that is close (nearby) whereas a route with a value of 15 is relatively far away. A route with a metric of 16 is considered to be unreachable (infinitely far away).

***SAME:** The value of the route metric will not change from its current value.

metric-value: Specify a metric value. Valid metric values range from 1 to 16.

REDST

Specifies whether this statically-defined route is to be redistributed (made available to other routers) in the future.

***SAME:** The route redistribution value will not change from its current setting.

***YES:** Specifies that this route will be shown (redistributed) to other routers.

***NO:** Specifies that this route will not be shown (redistributed) to other routers.

Note:

REDST(*YES) is analogous to the RIPv1 specification of STATIC. REDST(*NO) is analogous to the RIPv1 specification of PASSIVE.

DUPRTEPTY

The duplicate route priority value allows for ordering of duplicate routes within an internal route table. It allows specification of which particular duplicate route should be tried first in cases where a route cannot establish a connection.

***SAME:** The priority value will not change from its current value.

priority-value: Specify a priority value. Valid values are in the range of 1 to 10.

BINDIFC

The local IP interface to bind this route to. The binding is preferred and absolute.

If the IP interface defined for BINDIFC is active then the route specified will be bound to that interface.

***NONE:** TCP/IP will not attempt to bind this route to a particular IP interface but will bind it to the first active IP interface on the network as defined by the NEXTHOP and SUBNETMASK parameters.

binding-interface: Specify an IP interface to bind this route to. The binding is preferred and absolute.

Examples for CHGTCPRTE

Example 1: Changing a Route

```
CHGTCPRTE RTEDEST('132.65.0.0')
SUBNETMASK('255.255.0.0')
TOS(*MINDELAY) NEXTHOP('132.65.34.98')
MTU(1024)
```

This command changes the route identified by route destination 132.65.0.0 with a subnetmask of 255.255.0.0 and type of service of *MINDELAY. The change is to use a maximum transmission unit (MTU) of 1024.

Example 2: Changing a Default Route

```
CHGTCPRTE RTEDEST(*DFTRROUTE) SUBNETMASK(*NONE)
TOS(*NORMAL) NEXTHOP('186.49.126.108') MTU(1024)
```

This command changes the default route identified by next-hop value 186.49.126.108 to use an MTU value of 1024.

Error messages for CHGTCPRTE

*ESCAPE Messages

- TCP1D03**
&1 member record length not correct.
- TCP1D04**
Error occurred processing member &1 of &2/&3.
- TCP1901**
Internet address &1 not valid.
- TCP1902**
Internet address &1 not valid.
- TCP1908**
Internet address &1 not valid.
- TCP261C**
Process completed successfully.
- TCP2658**
&2 &1 not changed.
- TCP8050**
*IOSYSCFG authority required to use &1.
- TCP9509**
Line &1 not found.
- TCP9999**
Internal system error in program &1.

CHGTFTPA (Change TFTP Server Attributes) Command Description

CHGTFTPA Command syntax diagram

Purpose

The Change TFTP Server Attributes (CHGTFTPA) command is used to change the Trivial File Transfer Protocol (TFTP) server attributes. The changes take effect the next time the TFTP server is started either by the Start TCP/IP (STRTCP) command or by the Start TCP/IP Server (STRTCPSVR) command.

Restrictions:

You must have *IOSYSCFG special authority to use this command.

Optional Parameters

AUTOSTART

Specifies whether to automatically start the TFTP server when TCP/IP is started by the STRTCP command. When the TFTP server is started by the STRTCPSVR command, the AUTOSTART parameter is ignored and the TFTP mail server is started regardless of the value of this parameter. If STRTCPSVR *TFTP is specified, and the TFTP server is already running, then an additional server job is started.

***SAME:** The AUTOSTART value does not change if it was previously set. Otherwise, *NO is used.

***NO:** Do not start the number of server jobs defined in the NBRSVR parameter when the STRTCP command is called. If you do not intend to use the TFTP server, set AUTOSTART to *NO.

***YES:** Start the number of server jobs specified in the NBRSVR parameter.

ENBBCAST

Determines whether the subnet broadcast function of the TFTP Server is enabled. Allowing subnet broadcast can reduce load requirements on the TFTP server when simultaneously booting multiple

Network Stations from a particular subnet. Clients from the same subnet making compatible read requests including the subnet broadcast option can be broadcast the requested file. Compatible read requests are read requests for the same file, same mode, with a requested block size of greater than or equal to an existing broadcasting job's blocksize.

***SAME:** The value does not change if it was previously set. Otherwise, *YES is used.

***YES:** This enables the subnet broadcast function of the TFTP server.

***NO:** Disable the subnet broadcasting function of the TFTP server.

NBRSVR

The number of servers (NBRSVR) parameter has two parts, minimum and maximum.

Minimum specifies the number of TFTP server jobs to start when TFTP is started by either the Start TCP/IP (STRTCP) command or the Start TCP/IP Server (STRTCPSVR) command. These jobs allow new clients to connect to the server without having to wait for the overhead associated with starting a new job. The server tries to keep at least this number of jobs available for connecting to new clients as the number of connected clients changes. This is a performance enhancement for the TFTP server that reduces the system overhead each time a client connects.

The maximum is an upper limit on the number of TFTP server jobs.

Element 1: Minimum Number of Jobs

***SAME:** The number of server jobs previously set does not change. Otherwise two (2) is used.

***DFT:** The number of server jobs is set to the default value of 2.

minimum-number-of-server-jobs: Specify the number of server jobs to start. The valid range is 1 through 20.

Element 2: Maximum Number of Jobs

***SAME:** The number of server jobs previously set does not change. Otherwise six (6) is used.

***DFT:** The number of server jobs is set to the default value of 6.

maximum-number-of-server-jobs: Specify the maximum number of server jobs to start. The valid range is 1 through 250, but must be equal or greater than the minimum.

INACTTMR

During periods of inactivity, the number of active TFTP servers can drop to the minimum. The server inactivity timer (INACTTMR) specifies, in minutes, how often the primary TFTP server checks TFTP activity to see if a server can be terminated.

***SAME:** The server inactivity time value does not change if it was previously set. Otherwise 30 minutes is used.

***DFT:** The server inactivity time value is set to the default value of 30 minutes.

inactivity-timeout-value: Specify an server inactivity time value in the range 1 to 1440 minutes.

CCSID

Specifies the ASCII coded-character set identifier (CCSID) to use with Integrated File System. Integrated File System files will be read or write with this CCSID if they aren't in the "QIBM/ProdData" directory. Files in the "qibm/proddata" directory will be read in CCSID 00819.

***SAME:** The CCSID value that was previously set does not change. Otherwise, 00819 (ISO 8859-1 8-bit ASCII) is used.

***DFT:** The CCSID value is set to 00819 (ISO 8859-1 8-bit ASCII).

CCSID-value: Specify an ASCII CCSID value. This value is validated to ensure that you are specifying a valid ASCII CCSID.

MAXBLKSIZE

Specifies the maximum block size in bytes to send or receive data in. The supported values are 512 to 65464 bytes.

***SAME:** The block size doesn't change if it was previously set. Otherwise 1024 bytes is used.

***DFT:** The block size is set to 1024 bytes.

number-of-bytes: Specify the block size in number of bytes, for data transfer.

A block size of less than 1500 is suggested since most networks can support this packet size. Larger block sizes can increase performance but may not be compatible with certain mixes of routers and/or bridges.

RSPTIMO

Specifies the number of seconds to wait for an expected response before terminating the requested transfer. Re-transmissions may occur during this time period based on an internally calculated re-transmission timeout value.

***SAME:** The response timeout value does not change if it was previously set. Otherwise 60 seconds is used.

***DFT:** The response timeout value is set to the default value of 60 seconds.

response-timeout-value: Specify a response timeout value in the range 1 to 600 seconds.

ALWWRT

Allow file write (ALWWRT) determines whether TFTP users are allowed to add and update files on this system.

***SAME:** The value does not change if it was previously set. Otherwise, *NONE is used.

***NONE:** Do not allow TFTP users to 'PUT' data on this system.

***DFT:** The allow file write value is set to the default value of *NONE.

***CREATE:** Allow TFTP users to add new files to this system or overwrite old ones.

***REPLACE:** Allow TFTP users to overwrite existing files.

ALTSRCDIR

The alternate source directory specifies the integrated file system path to the directory whose files can be read.

***SAME:** The value that was previously set does not change. Otherwise, *NONE is used.

***NONE:** No alternate source directory is available. The source directory will be the special "/QIBM/ProdData" directory.

***DFT:** The path is set to *NONE.

path-to-alternate-source-dir: Specify the path to the source files.

Note that imbedded spaces and single quotes (apostrophe) will be removed.

ALTTGTDIR

The alternate target directory specifies the Integrated File System path to the directory to save files in.

***SAME:** The value that was previously set does not change. Otherwise, *NONE is used.

***NONE:** No alternate target directory is available. So, if write is enabled the target directory will be the special "/QIBM/ProdData/NetworkStation/Service" directory.

***DFT:** The path is set to *NONE.

path-to-alternate-target-directory: Specify the path to the target directory for files to be written.

Note that imbedded spaces and single quotes (apostrophe) will be removed.

Examples for CHGTFTP

Example 1: Start the TFTP server automatically

```
CHGTFTP AUTOSTART(*YES)
```

This command indicates that the next time the STRTCP command is issued to start up TCP/IP and to automatically start the TCP/IP applications, the TFTP server will be automatically started.

Example 2: Changing the number of initial server jobs

```
CHGTFTP NBRSVR(5)
```

This command indicates that the next time the TFTP server is started, five TFTP server jobs will be started automatically.

Example 3: Changing the number of server jobs

```
CHGTFTP NBRSVR(4 7)
```

This command indicates that the next time the TFTP server is started, four TFTP server jobs will be started automatically, and the maximum will be seven.

Error messages for CHGTFTP

*ESCAPE Messages

None.

CHGUSFDEVE (Change Ultimedia System Facilities Device Entry) Command Description

CHGUSFDEVE Command syntax diagram

Purpose

The Change Ultimedia System Facilities Device Entry (CHGUSFDEVE) command can be used to change an entry in the multimedia device table. The multimedia device entry describes the multimedia device.

Required Parameter

MMDEV

Specifies the multimedia device for which an entry is to be changed. The multimedia device is specified by the multimedia device name, the remote location name, and the remote network identifier (ID).

Element 1: Device Name

device-name: Specify the name of the multimedia device.

Element 2: Remote Location Name

***NONE**: The multimedia device does not have a remote location name.

remote-location-name: Specify the remote location name of the programmable work station (PWS) that is the multimedia device.

Element 3: Remote Network ID

***NONE**: The multimedia device does not have a remote network ID.

remote-network-identifier: Specify the network identifier of the remote system of the PWS that is the multimedia device.

Optional Parameters

DEVTYPE

Specifies the type of the multimedia device to be changed.

***SAME:** The value does not change.

***VIDEOTAPE:** The multimedia device is a video cassette recorder device.

***VIDEODISC:** The multimedia device is a videodisc recorder device.

***SWITCH:** The multimedia device is an audio switch or a combined video and audio switch.

***TUNER:** The multimedia device is a Personal System/2 television screen.

***CODEC:** The multimedia device is a videoconferencing combined coder and decoder device, which is used to compress and to decompress video images at the end of a phone line.

device-type: Specify the numeric value for the type of multimedia device.

EXITPGM

Specifies the name of the exit program used by the multimedia device.

***SAME:** The value does not change.

***ANCOR:** The exit program used is for an audio or video switch made by Ancor Communications, Inc.

***AUTOPATCH:** The exit program used is for an AutoPatch audio or video switch made by XN Technologies, Inc.

***JVC:** The exit program used is for a JVC** video cassette recorder made by Victor Company of Japan, Limited.

***NEC:** The exit program used is for a video cassette recorder made by NEC Technologies, Inc. or NEC Canada, Inc.

***NONCTBL:** The exit program used is for a multimedia device that is not controllable.

***PICTEL:** The exit program used is for a PictureTel** videoconferencing coder and decoder made by the PictureTel Corporation.

***P2200:** The exit program used is for a LaserDisc** videodisc recorder, model LD-V2200, made by the Pioneer Electronic Corporation.

***P8000:** The exit program used is for a LaserDisc videodisc recorder, model LD-V8000, made by the Pioneer Electronic Corporation.

***SIGMA:** The exit program used is for an audio or video switch made by Sigma Electronics Inc.

The name of the exit program can be qualified by a library value:

library-name: Specify the name of the library to be searched.

exit-program: Specify the name of the exit program.

DEVCTBL

Specifies whether the multimedia device can be controlled by the programmable work station (PWS). If the device is controllable, you can send instructions (such as play, stop, or rewind) over a communications line to control the multimedia device.

***SAME:** The value does not change.

***YES:** The multimedia device can be controlled.

***NO:** The multimedia device cannot be controlled.

SERVER

Specifies the name of server that controls the multimedia device.

***SAME:** The value does not change.

***NONE:** The multimedia device is not controlled by a server.

server-name: Specify the name of the server.

SVRPORT

Specifies the identifying number of the server port that is used to communicate with the multimedia device.

***SAME:** The value does not change.

***NONE:** No server port is specified. The multimedia device is not controlled by a server.

server-port-number: Specify the identifying number of the server port.

LINESPEED

Specifies the line speed in bits per second (bps).

***SAME:** The value does not change.

1200: The speed is 1200 bps.

2400: The speed is 2400 bps.

4800: The speed is 4800 bps.

9600: The speed is 9600 bps.

PARITY

Specifies the parity state that is used when communicating with the multimedia device.

***SAME:** The value does not change.

***NONE:** A parity state is not specified.

***EVEN:** The parity state is even-numbered.

***ODD:** The parity state is odd-numbered.

BITSCHAR

Specifies the number of data bits per character used when communicating with the multimedia device.

***SAME:** The value does not change.

8: The bits per character is 8 bits.

7: The bits per character is 7 bits.

STOPBITS

Specifies the number of stop bits that make up the stop signal used when communicating with the multimedia device.

***SAME:** The value does not change.

1: The number of stop bits is 2.

2: The number of stop bits is 1.

CAL

Specifies the name of the calendar that is associated with the multimedia device.

***SAME:** The value does not change.

***NONE:** No calendar is associated with the multimedia device.

calendar-name: Specify the name of the calendar.

TEXT Specifies the text that briefly describes the multimedia device. More information on this parameter is in Commonly used parameters.

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example for CHGUSFDEVE

```
CHGUSFDEVE MMDEV(EASYON)
TEXT('LOCAL VIDEO SWITCH')
```

This command changes the text for the device entry for the device EASYON. The text "LOCAL VIDEO SWITCH" becomes the description for the entry.

Error messages for CHGUSFDEVE

*ESCAPE Messages

None.

CHGUSRAUD (Change User Audit) Command Description

CHGUSRAUD Command syntax diagram

Purpose

The CHGUSRAUD (Change User Audit) command allows a user with *AUDIT special authority to set up or change auditing for a user. The system value QAUDCTL controls turning auditing on and off. The auditing attributes of a user profile can be displayed with the Display User Profile (DSPUSRPRF) command.

Note: The changes made by CHGUSRAUD take effect the next time a job is started for this user.

Required Parameter

USRPRF

Specifies the name of the user profile whose auditing values are changed.

Optional Parameters

OBJAUD

Specifies the object auditing value for the user. This value only takes effect if the object auditing (OBJAUD) value for the object being accessed has the value *USRPRF.

***SAME:** The value does not change.

***NONE:** The auditing value for the object determines when auditing is performed.

***CHANGE:** All change accesses by this user on all objects with the *USRPRF audit value are logged.

***ALL:** All change and read accesses by this user on all objects with the *USRPRF audit value are logged.

AUDLVL

Specifies the level of activity that is audited for this user profile.

Note:

The system value QAUDLVL is used in conjunction with this parameter. Example: If QAUDLVL is set to *DELETE and AUDLVL is set to *CREATE, then both *DELETE and *CREATE would be audited for this user. The default value for the QAUDLVL system value is *NONE.

***SAME:** The value does not change.

***NONE:** No auditing level is specified. The auditing level for this user is taken from system value QAUDLVL.

***CMD:** CL command strings, System/36 environment operator control commands, and System/36 environment procedures are logged for this user.

***CREATE:** Auditing entries are sent when objects are created by this user.

***DELETE:** Auditing entries are sent when objects are deleted by this user.

***JOBDTA:** All job start and stop data is audited for this user.

***OBJMGT:** Object management changes made by this user, such as move or rename, are audited.

***OFCSRVR:** Office services changes made by this user, such as changes to the system directory and use of OfficeVision mail, are audited.

***OPTICAL:** Optical actions performed by this user are audited.

***PGMADP:** Authority obtained through program adoption is audited for this user.

***SAVRST:** Save and restore actions performed by this user are audited.

***SECURITY:** Security changes made by this user are audited.

***SERVICE:** Use of the system service tools by this user is audited.

***SPLFDTA:** Spool files operations made by this user are audited.

***SYSMGT:** Use of system management functions by this user is audited.

Example for CHGUSRAUD

```
CHGUSRAUD  USRPRF(FRED)  OBJAUD(*CHANGE)
           AUDLVL(*CREATE *DELETE)
```

This command changes the auditing value in the user profile of the user FRED. All objects whose object auditing value is *USRPRF are audited when they are changed by user FRED. All objects that are created and all objects that are deleted will be audited for user FRED. Auditing records are sent to the auditing journal QAUDJRN in QSYS.

Error messages for CHGUSRAUD

***ESCAPE Messages**

CPF22B0

Not authorized to change the auditing value.

CPF22CC

Auditing value not changed for some user profiles.

CPF22FF

System user profile cannot be changed.

CHGUSRPRTI (Change User Print Information) Command Description

CHGUSRPRTI Command syntax diagram

Purpose

The Change User Print Information (CHGUSRPRTI) command changes the user print information for a particular user by altering the user defined text value within the system.

Optional Parameters

USER Specifies the name of the user whose print information is being changed.

***CURRENT:** The user profile that is currently running is used.

user-name: Specify the name of the user whose print information is being changed.

TEXT Specifies the text that briefly describes the print information. More information on this parameter is in Commonly used parameters.

This text is retrieved for the current user when spooled files are created and can be displayed using the Work with Spooled File Attributes (WRKSPLFA) command.

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify a maximum of 100 characters of text to describe the user print information.

Example for CHGUSRPRTI

```
CHGUSRPRTI  USER(FEIST)
             TEXT('DEPT. 456 P.O. BOX 123')
```

This command changes the user print information for user profile FEIST. The user print information is changed to "DEPT. 456 P.O. BOX 123."

Error messages for CHGUSRPRTI

***ESCAPE Messages**

CPF0011

Error detected by prompt override program.

CPF2204

User profile &1 not found.

CPF2213

Not able to allocate user profile &1.

CPF2217

Not authorized to user profile &1.

CPF2225

Not able to allocate internal system object.

CPF2247

Internal security object not available. Reason code &1.

CPF34D2

User print information not changed for user &1.

CPF34D5


CCSID translation error.

CHGUSRPRF (Change User Profile) Command Description

CHGUSRPRF Command syntax diagram

Purpose

The Change User Profile (CHGUSRPRF) command changes the values specified in a user profile.

The password validation rules are not verified by the system when a password is changed by this command. A description of the password validation rules is in the iSeries Security Reference  book.

Restrictions:

1. You must have *SECADM special authority, and *OBJMGT and *USE authorities to the user profile being changed, can specify this command.
2. *USE authority to the current library, program, menu, job description, message queue, print device, output queue, or ATTN key handling program is required to specify these parameters.

Required Parameter

USRPRF

Specifies the name of the user profile whose values are being changed. A numeric user profile can be specified. If the user profile is numeric, then it must begin with a Q.

The following IBM-supplied objects are not valid on this parameter: 

QAUTPROF	QDSNX	QPEX	QYPSJSVR
QCOLSRV	QFNC	QPM400	
QDBSHR	QGATE	QSNADS	
QDBSHRDO	QIPP	QSPL	
QDFTOWN	QLPAUTO	QSPLJOB	
QDIRSRV	QLPINSTALL	QSYS	
QDLFM	QMSF	QTCP	
QDOC	QNTF	QTSTRQS	




Optional Parameters

PASSWORD

Specifies the password that allows the user to sign on the system. The password is associated with a user profile and is used by the system to represent the user in the system. The passwords should be known only to the individual user. A numeric password can be specified. When the system is operating at password level 0 or 1 and the password is numeric, then the password must begin with a Q, for example, Q1234 where 1234 is the password used for signing on the system.

Note: The password level is controlled by the Password Level (QPWDLVL) system value.

Note: The new password is not checked against the password validation rules. The password validation rules are defined by OS/400 system values. For a description of the password validation rules, see the iSeries Security Reference  book.

***SAME:** The value does not change.

***NONE:** No password is associated with this user profile. Users cannot sign on a system with a profile that has PASSWORD(*NONE) specified.

'user-password': When the system is operating at password level 0 or 1, specify an alphanumeric character string of 10 characters or less. The first character must be alphabetic and the other characters must be alphanumeric.

When the system is operating at password level 2 or 3, specify a character string of 128 characters or less.

SPCAUT

Specifies the special authorities given to a user. Special authorities are required to perform certain functions on the system. Special authorities cannot be removed from many of system supplied user profiles including QSECOFR and QSYS.

Restrictions:

1. The user profile creating or changing another user profile must have all of the special authorities being given. All special authorities are needed to grant all special authorities to another user profile.
2. A user must have *ALLOBJ and *SECADM special authorities to grant a user *SECADM special authority when using the CHGUSRPRF command.
3. The user must have *ALLOBJ, *SECADM, and *AUDIT special authorities to grant a user *AUDIT special authority when using the CHGUSRPRF command.

Specify one of the following values:

***SAME:** The value does not change.

***USRCLS:** Special authorities are given to the user based on the value entered in the USRCLS parameter.

***NONE:** No special authority is given to the user.

Or specify one or more of the following:

***SAVSYS:** Save system authority is given to the user profile. This user has the authority to save, restore, and free storage for all objects on the system, with or without object management authority.

***IOSYSCFG:** Input/output (I/O) system configuration authority is granted to the user. The user has authority to change system I/O configurations.

***JOBCTL:** Job control authority is given to the user. The user has authority to change, display, hold, release, cancel, and clear all jobs that are running on the system or that are on a job queue or output queue that has OPRCTL (*YES) specified. The user also has the authority to start writers and to stop active subsystems.

***ALLOBJ:** All object authority is given to the user. The user can access any system resource with or without private user authorizations.

***SECADM:** Security administrator authority is given to the user. The user can create, change, or delete user profiles if authorized to the Create User Profile, Change User Profile, or Delete User Profile commands and is authorized to the user profile. This right does not allow giving special authorities not in this user's profile.

***SERVICE:** Service authority is given to the user. The user can perform service functions.

***SPLCTL:** Spool control authority is given to the user. The user can perform all spool functions.

***AUDIT:** Audit authority is granted to the user. This user is given the authority to perform auditing functions. Auditing functions include turning auditing on or off for the system and controlling the level of auditing on an object or user.

PWDEXP

Specifies that the user's password is set to expired. If the password is set to expired, the user is required to change the password to sign on the system. When the user attempts to sign on the system, the Sign-On Information display is shown and the user has the option to change the password.

***SAME:** The value does not change.

***NO:** The password is not set to expired.

***YES:** The password is set to expired.

STATUS

Specifies whether the profile is in an enabled or disabled status.

***SAME:** The value does not change.

***ENABLED:** The profile is enabled.

***DISABLED:** The profile is disabled.

USRCLS

Specifies the class of user associated with this user profile: security officer, security administrator, programmer, system operator, or user. User class determines which menu options are shown. The special authorities defined by the user class are used only if SPCAUT(*USRCLS) is specified. If SPCAUT(*USRCLS) is specified, the special authorities granted will differ depending on the QSECURITY value.

***SAME:** The value does not change.

***SECOFR:** At all levels of security, the security officer has the following special authorities:

*ALLOBJ

*SAVSYS

*JOBCTL

*SERVICE

*SPLCTL

*SECADM

*AUDIT

*IOSYSCFG

***SECADM:** At QSECURITY level 10 or 20, the security administrator has *ALLOBJ, *SAVSYS, *SECADM, and *JOBCTL special authorities.

At QSECURITY level 30 or above, the user has *SECADM special authority.

***PGMR:** At QSECURITY level 10 or 20, the programmer has *ALLOBJ, *SAVSYS, and *JOBCTL special authorities.

At QSECURITY level 30 or above, the user has no special authorities.

***SYSOPR:** At QSECURITY level 10 or 20, the system operator has *ALLOBJ, *SAVSYS, and *JOBCTL special authorities.

At QSECURITY level 30 or above, the user has *SAVSYS and *JOBCTL special authorities.

***USER:** At QSECURITY level 10 or 20, the user has *ALLOBJ and *SAVSYS authority.

At QSECURITY level 30 or above, the user has no special authorities.

ASTLVL

Specifies which user interface is used.

***SAME:** The value does not change.

***SYSVAL:** The system determines the graphic character set and code page values for the command parameters from the QCHRID system values.

***BASIC:** The Operational Assistant* user interface is used.

***INTERMED:** The system interface is used.

***ADVANCED:** The expert system interface is used. To allow for more list entries, the options keys and the function keys are not displayed. If a command does not have an advanced (*ADVANCED) level, the intermediate (*INTERMED) level is used.

SPCENV

Specifies the special environment in which the user operates after signing on the system.

***SAME:** The value does not change.

***SYSVAL:** The system value, QSPCENV, is used to determine the system environment after the user signs on the system.

***NONE:** The user operates in the iSeries 400 environment after signing on the system.

***S36:** The user operates in the System/36 environment after signing on the system.

DSPSGNINF

Specifies whether the sign-on information display is shown when the user signs on the system. This allows users to see the sign-on information, such as date of last sign-on and sign-on attempts that were not valid. If the password is due to expire in 7 days or less, the number of days until the password expires is shown.

***SAME:** The value does not change.

***SYSVAL:** The system value QDSPSGNINF is used to determine whether the sign-on information display is shown when the user signs on the system.

***NO:** The sign-on information display is not shown when the user signs on the system.

***YES:** The sign-on information display is shown when the user signs on the system.

PWDEXPITV

Specifies the number of days (from the password changed date) before a password expires.

***SAME:** The value does not change.

***SYSVAL:** The system value QPWDEXPITV is used to determine the password expiration interval.

***NOMAX:** There is no password expiration interval.

password-expiration-interval: Specify the number of days after which the password expires. Valid values range from 1 to 366.

LMTDEVSSN

Specifies whether the number of device sessions allowed for the user is limited to one. This does not limit the System Request menu or a second sign-on.

***SAME:** The value does not change.

***SYSVAL:** The system value QLMTDEVSSN is used to determine whether the user is limited to one device session.

***NO:** The user is not limited to one device session.

***YES:** The user is limited to one device session.

KBDBUF

Specifies the keyboard buffering value used when a job is initialized for this user profile. The new value takes effect the next time the user signs on. If the type-ahead feature is active, the keystrokes can be buffered. If the attention key buffering option is active, the attention key is buffered like any other key. If the attention key is not active, the attention key is not buffered and is sent to the system even if the display station is input inhibited. The keyboard buffer value can also be set by a user application using the QWSSETWS program. More information is in Application Program Interfaces (APIs) topic in the Information Center.

***SAME:** The value does not change.

***SYSVAL:** The system value, QKBDBUF, is used to determine the keyboard buffering value for this profile.

***NO:** The type-ahead feature and attention key buffering option are not active for this user profile.

***TYPEAHEAD:** The type-ahead feature is active for this user profile.

***YES:** The type-ahead feature and attention key buffering option are active for this user profile.

MAXSTG

Specifies the maximum amount of auxiliary storage (kilobytes) assigned to store permanent objects owned by this user profile (1 kilobyte equals 1024 bytes). If the maximum is exceeded when an interactive user tries to create an object, an error message is displayed, and the object is not created. If the maximum is exceeded when an object is created in a batch job, an error message is sent to the job log (depending on the logging level of the job), and the object is not created.

Storage is allocated in 4K increments. Therefore, if you specify MAXSTG (9), the profile is allocated 12K of storage.

When planning maximum storage for user profiles, consider the following system actions:

- A restore operation assigns the storage to the user doing the restore, and then transfers the object to the owner. For a large restore, specify MAXSTG(*NOMAX).
- The user profile that creates a journal receiver is assigned the required storage as the receiver size grows. If new receivers are created using JRNRCV(*GEN), the storage continues to be assigned to the user profile that owns the active journal receiver. If a very active journal receiver is owned, specify MAXSTG(*NOMAX).

- User profiles that transfer created objects to their group profile must have adequate storage in the user profiles to contain created objects before the objects are transferred to the group profile.
- The owner of the library is assigned the storage for the descriptions of objects which are stored in a library, even when the objects are owned by another user profile. Examples of such objects are text and program references.
- Storage is assigned to the user profile for temporary objects used while a job is running. Examples of such objects are commit control blocks, file editing space, and documents.

***SAME:** The value does not change.

***NOMAX:** As much storage as required is assigned to this profile.

maximum-K-bytes: Specify the maximum amount of storage (in kilobytes) that can be assigned to the user with this profile (1 kilobyte equals 1024 bytes). The maximum storage allowed can be less than the current maximum storage used for the user profile.

PTYLMT

Specifies the highest scheduling priority the user is allowed to have for each job submitted to the system. This value controls the job processing priority and output priority for any job running under this user profile; that is, values specified in the JOBPTY and OUTPTY parameters of any job command cannot exceed the PTYLMT value of the user profile under which the job is run. The scheduling priority can have a value ranging from 1 through 9, where 1 is the highest priority and 9 is the lowest priority. More information on this parameter is in Commonly used parameters.

***SAME:** The value does not change.

priority-limit: Specify a value ranging from 1 to 9 for the highest scheduling priority allowed to the user.

CURLIB

Specifies the name of the library being used as the current library for jobs initiated by this user profile. If *PARTIAL or *YES is specified for the LMTCPB parameter in the user profile, the user cannot change the current library at sign-on or by using the Change Profile (CHGPRF) command.

***SAME:** The value does not change.

***CRTDFT:** This user has no current library. If objects are created and placed in the current library by using *CURLIB on a create command, the QGPL library is used as the default current library.

current-library-name: Specify the name of the library that is this user's current library after sign-on to the system.

INLPGM

Specifies, for an interactive job, the name of the program called whenever a new routing step is started that has QCMD as the request processing program. If *PARTIAL or *YES is specified on the LMTCPB parameter in the user profile, the program value cannot be changed at sign on or by using the Change Profile (CHGPRF) command. No parameters can be passed to the program.

A system/36 environment procedure name can be specified as the initial program if the procedure is a member of the file QS36PRC (in the library list or specified library) and if either of the following conditions are true:

- *S36 is specified for the SPCENV parameter.
- *SYSVAL is specified for the SPCENV parameter and the system value, QSPCENV, is *S36.

***SAME:** The value does not change.

***NONE:** No program is called when the user signs on the system. If a menu name is specified in the INLMNU parameter, that menu is displayed.

The name of the program 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.

program-name: Specify the name of the program called after the user signs on the system.

INLMNU

Specifies the name of the initial menu displayed when the user signs on the system if the user's routing program is the command processor QCMD. If *YES is specified for the LMTCPB parameter in the user profile, the user cannot change the menu either at sign-on or with the Change Profile (CHGPRF) command.

A system/36 environment menu can be specified as the initial menu if either of the following conditions are true:

- *S36 is specified for the SPCENV parameter.
- *SYSVAL is specified for the SPCENV parameter and the system value, QSPCENV, is *S36.

***SAME:** The value does not change.

***SIGNOFF:** The system signs-off the user when the initial program completes. This is intended for users authorized only to run the program.

The name of the menu 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.

menu-name: Specify the name of the initial menu called after the user signs on the system.

LMTCPB

Specifies the limit to which the user can control the program, menu, current library, and the ATTN key handling program values. It also determines whether the user can run commands from a command line. This parameter is ignored when the security level is 10.

Note:

When creating or changing other users' user profiles, users running this command cannot specify values on this parameter that grant greater capabilities to other users than their own user profiles grant to them. For example, if *PARTIAL is specified on the LMTCPB parameter in the user profile of the user running this command, *PARTIAL or *YES can be specified for another user. *NO cannot be specified for another user.

***SAME:** The value does not change.

***NO:** The program, menu, and current library values can be changed when the user signs on the system. Users may change the program, menu, current library, or ATTN key handling program values in their own user profiles with the Change Profile (CHGPRF) command. Commands can be run from a command line.

***PARTIAL:** The program and current library cannot be changed on the sign-on display. The menu can be changed and commands can be run from a command line. A user can change the menu value with the Change Profile (CHGPRF) command. The program, current library, and the ATTN key handling program cannot be changed using the Change Profile command.

***YES:** The program, menu, and current library values cannot be changed on the sign-on display. Commands cannot be run when issued from a command line or by selecting an option from a command grouping menu such as CMDADD, but can still be run from a command entry screen. The user cannot change the program, menu, current library, or the ATTN key program handling values by using the Change Profile (CHGPRF) command.

JOB Specifies the name of the job description used for jobs that start through subsystem work station entries. If the job description does not exist when the user profile is created or changed, a library qualifier must be specified, because the job description name is kept in the user profile.

***SAME:** The job description name does not change.

The name of the job description 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.

job-description-name: Specify the name of the job description used for the work station entries whose job description parameter values indicate the user (JOB(*USRPRF)).

GRPPRF

Specifies the user's group profile name whose authority is used if no specific authority is given for the user. The current user of this command must have *OBJMGT, *CHANGE, *OBJOPR, *READ, *ADD, *UPD, AND *DLT authority to the profile specified on the GRPPRF parameter. The required *OBJMGT authority cannot be given by a program adopt operation.

Notes:

1. When a group profile is specified, the user is automatically granted *CHANGE and *OBJMGT authority to the group profile.
2. The following IBM-supplied objects are not valid on this parameter: ➤

QAUTPROF	QFNC	QSNADS
QCLUMGT	QGATE	QSPL
QCLUSTER	QIPP	QSPLJOB
QCOLSRV	QLPAUTO	QSRV
QDBSHR	QLPINSTALL	QSRVBAS
QDBSHRDO	QMSF	QSYS
QDFTOWN	QNETSPLF	QTCM
QDIRSRV	QNFSANON	QTCP
QDLFM	QNTF	QTFTP
QDOC	QPEX	QTSTRQS
QDSNX	QPM400	QYPSJSVR
QEJB	QRJE	



***SAME:** The value does not change.

***NONE:** This user profile has no group profile.

user-profile-name: Specify the name of the group profile used with this user profile.

OWNER

Specifies the user profile that is the owner of newly created objects.

***SAME:** The value does not change.

***USRPRF:** The user profile associated with the job is the owner of the object.

***GRPPRF:** The group profile is made the owner of newly created objects and has all authority to the object. The user profile associated with the job does not have any specific authority to the object. If *GRPPRF is specified, a user profile name must be in the GRPPRF parameter, and the GRPAUT parameter cannot be specified.

GRPAUT

Specifies the specific authority given to the group profile for newly created objects. If *GRPPRF is specified on the OWNER parameter, specification of this parameter is not allowed.

***SAME:** The value does not change.

***NONE:** No group authority is granted.

***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 also can change ownership of the object.

***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 change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***USE:** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. *USE authority provides object operational authority, read authority, and execute authority.

***EXCLUDE:** The user cannot access the object.

GRPAUTYP

Specifies the type of authority to be granted to the group profile for newly-created objects. If

*NONE is specified on the GRPAUT parameter, specification of this parameter is ignored.

***SAME:** The value does not change.

***PRIVATE:** The group profile is granted private authority to newly-created objects, with the authority value determined by the GRPAUT parameter. If the authority value in the GRPAUT parameter is *NONE, this value is ignored.

***PGP:** The group profile is the primary group for newly-created objects, with the authority value determined by the GRPAUT parameter. If the authority value in the GRPAUT parameter is *NONE, this value is ignored.

SUPGRPPRF

Specifies the user's supplemental group profiles. The profiles specified here, along with the group profile specified on the GRPPRF parameter, are used to determine what authority the user has if no specific user authority is given for the job. If profiles are specified for this parameter, a group profile name must have been specified on the GRPPRF parameter for this user profile (either on this command or a previous Create User Profile (CRTUSRPRF) or CHGUSRPRF command. The current user of this command must have *OBJMGT, *CHANGE, *OBJOPR, *READ, *ADD, *UPD, AND *DLT authority to the profiles specified on the SUPGRPPRF parameter. The required *OBJMGT authority cannot be given by a program adopt operation.

Notes:

1. When a group profile is specified, the user is automatically granted *CHANGE, *OBJMGT, *OBJOPR, *READ, *ADD, *UPD, AND *DLT authority to the group profile.
2. The following IBM-supplied objects are not valid on this parameter: ➤

QAUTPROF	QFNC	QSNADS
QCLUMGT	QGATE	QSPL
QCLUSTER	QIPP	QSPLJOB
QCOLSRV	QLPAUTO	QSRV
QDBSHR	QLPINSTALL	QSRVBAS
QDBSHRDO	QMSF	QSYS
QDFTOWN	QNETSPLF	QTCM
QDIRSRV	QNFSANON	QTCP
QDLFM	QNTF	QTFTP
QDOC	QPEX	QTSTRQS
QDSNX	QPM400	QYPSJSVR
QEJB	QRJE	



***SAME:** The value does not change.

***NONE:** No supplemental group profiles are used with this user profile.

group-profile-name: Specify a maximum of 15 group profile names used with this user profile and the group profile specified on the GRPPRF parameter to determine a job's eligibility for getting access to existing objects and special authority.

ACGCDE

Specifies the accounting code associated with this user profile. More information on job accounting is in the Database Programming topic in the Information Center.

***SAME:** The value does not change.

***BLANK:** An accounting code consisting of 15 blanks is assigned to this user profile.

accounting-code: Specify the 15-character accounting code to be used by jobs that get their accounting code from this user profile. If less than 15 characters are specified, the string is padded on the right with blanks.

DOCPWD

Specifies the document password that allows Document Interchange Architecture (DIA) users to protect personal distributions from being used by users working on their behalf.

***SAME:** The value does not change.

***NONE:** No document password is used by this user.

document-password: Specify a document password for use by this user. The password must consist of alphanumeric characters ranging from 1 through 8 (letters A through Z and numbers 0 through 9). The first character of the document password must be alphabetic; the remaining characters can be alphanumeric. Embedded blanks, leading blanks, and special characters are not valid.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

Note:

The message queue is created if it does not already exist. The user profile being changed is the owner of the message queue.

***SAME:** The value does not change.

***USRPRF:** A message queue with the same name as that specified in the USRPRF parameter is used as the message queue for this user. QUSRSYS is the library where this message queue is located.

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, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue used by this user.

DLVRY

Specifies how the messages sent to the message queue for the user profile are delivered.

***SAME:** The value does not change.

***HOLD:** The messages are held in the message queue until they are requested by the user or program.

***BREAK:** The job, to which the message queue is assigned, is interrupted when a message reaches the message queue. If the job is an interactive job, the audible alarm is sounded (if the alarm is set). The delivery mode cannot be changed to *BREAK if the message queue is also being used by another job.

***NOTIFY:** The job to which the message queue is assigned is notified when a message reaches the message queue. For interactive jobs at a work station, the audible alarm is sounded (if the alarm is set) and the message-waiting light is turned on. The delivery mode cannot be changed to *NOTIFY if the message queue is also being used by another job.

***DFT:** The default reply to the inquiry message is sent. If no default reply is specified in the message description of the inquiry message, the system default reply, *N, is used.

SEV Specifies the lowest severity code that a message can have and still be delivered to a user in break or notify mode. Messages arriving at the message queue whose severities are lower than the severity code specified on this parameter do not interrupt the job or turn on the audible alarm or the message-waiting light; they are held in the queue until they are requested by using the Display Message (DSPMSG) command. If *BREAK or *NOTIFY is specified on the DLVRY parameter, and is in effect when a message arrives at the queue, the message is delivered if the severity code associated with the message is equal to or greater than the value specified here. Otherwise, the message is held in the queue until it is requested.

***SAME:** The value does not change.

severity-code: Specify severity code ranging from 00 through 99.

PRTDEV

Specifies the name of the default printer device for this user. If the printer file being used to create the output specifies to spool the file, the spooled file is placed on the device's output queue, which is named the same as the device.

Note:

This assumes the defaults are specified on the OUTQ parameter for the printer file, job description, user profile and workstation.

***SAME:** The value does not change.

***WRKSTN:** The printer assigned to the user's work station is used.

***SYSVAL:** The value specified in the system value QPRTDEV is used.

print-device-name: Specify the name of a printer used to print the output for this user.

OUTQ Specifies the qualified name of the output queue.

***SAME:** The value does not change.

***WRKSTN:** The output queue assigned to the user's work station is used.

***DEV:** The output queue associated with the printer specified on the DEV parameter is used. The output queue has the same name as the printer. (The printer file DEV parameter is determined by the Create Printer File (CRTPRTF), Change Printer File (CHGPRTF), or Override with Printer File (OVRPRTF) command.)

Note:

This assumes the defaults were specified on the OUTQ parameter for the printer file, job description, user profile, and workstation.

The name of the output 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, the QGPL library is used.

library-name: Specify the name of the library to be searched.

output-queue-name: Specify the name of the output queue used as the output queue for this user.

ATNPGM

Specifies the program to be the ATTN key handling program for this user. The ATTN key handling program is called when the ATTN key is pressed during an interactive job. The program is active only when the user specifies routes to the system-supplied QCMD command processor. The ATTN key handling program is activated before the program (if any) is called and is active for both program and menu. If the program changes the ATNPGM (by using the Set Attention Program (SETATNPGM) command), the new program remains active only for the duration of the program. When control returns and QCMD calls the menu, the original ATTN key handling program becomes active again. If the SETATNPGM command is run from the menus or an application is called from the menus, the new ATTN key handling program specified overrides the original ATTN key handling program. If *YES or *PARTIAL is specified on the LMTCPB parameter in the user profile, the ATTN key handling program cannot be changed by the Change Profile (CHGPRF) command.

***SAME:** The value does not change.

***ASSIST:** QEZMAIN is used.

***SYSVAL:** The system value QATNPGM is used.

***NONE:** No ATTN key handling program is used by this user.

The name of the ATTN key handling program 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.

program-name: Specify the name of the ATTN key handling program used by this user.

SRTSEQ

Specifies the sort sequence table to be used for string comparisons for this user profile.

***SAME:** This value does not change.

***SYSVAL:** The system value QSRTSEQ is used.

***HEX:** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

***LANGIDUNQ:** A unique-weight sort table is used.

***LANGIDSHR:** A shared-weight sort table is used.

The name of the sort sequence table 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.

table-name: Specify the name of the sort sequence table to be used with this user profile.

LANGID

Specifies the language identifier used for this user.

***SAME:** The language identifier does not change.

***SYSVAL:** The system value QLANGID is used.

language-identifier: Specify the language identifier. More information on valid language identifiers is in the Globalization topic in the Information Center.

CNTRYID

Specifies the country or region identifier to be used by the system for this user.

***SAME:** The country or region identifier does not change.

***SYSVAL:** The system value QCNTRYID is used.

country-identifier: Specify an ISO 3166 Alpha-2 code from the country or region code table. More information on this parameter is in Commonly used parameters.

CCSID

Specifies the coded character set identifier (CCSID) used for this user.

A CCSID is a 16-bit number identifying a specific set of encoding scheme identifiers, character set identifiers, code page identifiers, and additional coding-related information that uniquely identifies the coded graphic representation used.

Note: If the value for CCSID is changed, the change does not affect jobs that are currently running.

***SAME:** The CCSID does not change.

***SYSVAL:** The system value QCCSID is used.

***HEX:** The CCSID 65535 is used.

coded-character-set-identifier: Specify the CCSID. More information on valid CCSIDs is in the Globalization topic in the Information Center.

CHRIDCTL

Specifies the character identifier control for the job. This attribute controls the type of CCSID conversion that occurs for display files, printer files and panel groups. The *CHRIDCTL special value must be specified on the CHRID command parameter on the create, change or override commands for display files, printer files and panel groups before this attribute will be used.

***SAME:** The CHRID control does not change.

***SYSVAL:** The value in the QCHRIDCTL system value will be used.

***DEVVD:** The *DEVVD special value performs the same function as on the CHRID command parameter for display files, printer files and panel groups.

***JOBCCSID:** The *JOBCCSID special value performs the same function as on the CHRID command parameter for display files, printer files and panel groups.

SETJOBATR

Specifies which job attributes are taken from the locale specified in the LOCALE parameter when the job is initiated.

***SAME:** The value does not change.

***NONE:** No job attributes are taken from the locale.

***SYSVAL:** The system value QSETJOBATR is used to determine which job attributes are taken from the locale.

Any combination of the following values can be specified:

***CCSID:** The coded character set identifier from the locale is used. The CCSID value from the locale overrides the user profile CCSID.

***DATFMT:** The date format from the locale is used.

***DATSEP:** The date separator from the locale is used.

***DECfmt:** The decimal format from the locale is used.

***SRTSEQ:** The sort sequence from the locale is used. The sort sequence from the locale overrides the user profile sort sequence.

***TIMSEP:** The time separator from the locale is used.

USROPT

Specifies the level of information detail the user can view and the function of the Page Up and Page Down keys by default. The system shows displays suitable for the inexperienced user. More experienced users must perform an extra action to see more detailed information. When values are specified for this parameter, the system presents detailed information without additional action by the user.

***SAME:** The value does not change.

***NONE:** No detailed user option information is shown.

***CLKWD:** Parameter keywords are shown instead of the possible parameter values when a command is displayed.

***EXPERT:** More detailed user option information is initially shown when the user is performing display and edit options (such as edit or display object authority) to define or change the system.

***ROLLKEY:** The actions of the Page Up and Page Down keys are reversed.

***HLPFULL:** Help text is shown on a full display rather than in a window.

***NOSTMSG:** Status messages are not displayed when sent to the user.

***STMSG:** Status messages are displayed when sent to the user.

***PRTMSG:** A message is sent to this user's message queue when a spooled file for this user is printed or held by the printer writer.

UID

Specifies the user ID number (uid number) for this user profile. The uid number is used to identify the user when the user is using the directory file system. The uid number for a user cannot be changed if the user is active in a process.

***SAME:** The value does not change.

user-ID-number: Specify the uid number to be assigned to the user profile. Valid values range from 1 to 4294967294. The uid number assigned must not already be assigned to another user profile.

GID

Specifies the group ID number (gid number) for this user profile. The gid number is used to identify the group profile when a member of the group is using the directory file system. The gid number for a user cannot be changed if:

- The user profile is the primary group of an object in a directory.

- The user is active in a process.

***SAME:** The value does not change.

***NONE:** The user does not have a gid number or an existing gid number is removed.

Note: This value cannot be specified if the user is a group profile or the primary group of an object.

***GEN:** The gid number is generated for the user. The system generates a gid number that is not already assigned to another user. The gid number generated is greater than 100.

group-ID-number: Specify the gid number to be assigned to the user profile. Valid values range from 1 to 4294967294. The gid number assigned must not already be assigned to another user profile.

HOMEDIR

Specifies the path name of the home directory for this user profile. The home directory is the user's initial working directory. The working directory, associated with a process, is used during path name resolution in the directory file system for path names that do not begin with a slash (/). If the home directory specified does not exist when the user signs on, the user's initial working directory is the root (/) directory.

Note: This parameter is only used with the integrated file system. It cannot be used to set the home directory for IBM OS/2 Warp Server for AS/400.

***SAME:** The value does not change.

***USRPRF:** The home directory assigned to the user is /home/USRPRF, where USRPRF is the name of the user profile.

'home-directory-path-name': Specify the path name of the home directory to assign to this user. See path names for more information on specifying path names.

LOCALE

Specifies the path name of the locale that is assigned to the LANG environment variable for this user.

***SAME:** The value does not change.

***SYSVAL:** The system value QLOCALE is used to determine the locale path name assigned to this user.

***NONE:** No locale path name is assigned to this user.

***C:** The C locale path name is assigned to this user.

***POSIX:** The POSIX locale path name is assigned to this user.

'locale path name': Specify the path name of the locale assigned to this user. See path names for more information on specifying path names.

TEXT Specifies the text that briefly describes the user profile name specified in the USRPRF parameter. More information on this parameter is in Commonly used parameters.

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example for CHGUSRPRF

```
CHGUSRPRF USRPRF(JJADAMS) PASSWORD(SECRET)
          SPCAUT(*JOBCTL) INLPGM(ARLIB/DSPMENU)
```

This command makes the following changes to the user profile named JJADAMS:

- Changes the password to SECRET.
- Authorizes JJADAMS to use the special job control authority.
- Changes the first program to start following a successful sign-on to a program named DSPMENU, which is located in a library named ARLIB.

All the other command parameters default to *SAME and do not change.

Error messages for CHGUSRPRF

*ESCAPE Messages

CPF22CD

Value for SUPGRPPRF parameter is not correct.

CPF22CE

The &1 value &2 is used by another user profile.

CPF22CF

User profile not allowed to be a group profile.

CPF22DB

The user profile being changed must have a GID.

CPF22DC

Not allowed to change UID of the user profile.

CPF22DD

Not allowed to change GID of the user profile.

CPF22DE

Not allowed to change the UID or GID of user profile &1.

CPF22DF

Unable to process request for user profile &1.

CPF22EB

Unable to process request for user profile &1.

CPF22E1

USROPT parameter cannot specify *STSMMSG and *NOSTSMMSG.

CPF22F1

Coded character set identifier &1 not valid.

CPF22F3

&1 specified a LMTCPB value that is not permitted.

CPF2203

User profile &1 not correct.

CPF2204

User profile &1 not found.

CPF2209

Library &1 not found.

CPF2213
Not able to allocate user profile &1.

CPF2225
Not able to allocate internal system object.

CPF2228
Not authorized to change user profile.

CPF223F
Cannot set password to expired when password is *NONE.

CPF224A
User profile &1 cannot have a GID and be a member of a group.

CPF2242
Object &1 type *&2 not found in library list.

CPF2244
Object &1 type *&2 cannot be found.

CPF225A
User profile name specified on both USRPRF and SUPGRPPRF parameters.

CPF2259
Group profile &1 not found.

CPF2260
User profile &2 was not created or changed. Reason code &3.

CPF2261
OWNER or GRPAUT value not permitted.

CPF2262
Value for GRPAUT not correct.

CPF2264
User profile &1 not allowed to be a group member.

CPF2269
Special authority *ALLOBJ required when granting *SECADM or *AUDIT.

CPF2272
Cannot allocate user profile &1.

CPF2291
User profile does not have all special authorities being granted.

CPF2292
*SECADM required to create or change user profiles.

CPF2293
Storage limit exceeded for user profile &1.

CPF9802
Not authorized to object &2 in &3.

CPF9820
Not authorized to use library &1.

CPF9825
Not authorized to device &1.

CHGUSRTRC (Change User Trace Buffer) Command Description

CHGUSRTRC Command syntax diagram

Purpose

The Change User Trace Buffer (CHGUSRTRC) command changes the user trace buffer associated with the specified job. Each user trace buffer is a user space (*USRSPC) object in library QUSRSYS by the name QP0Znnnnnn, where 'nnnnnn' is the job number of the job using the user trace facility.

The user trace facility supports user-generated tracepoints written by way of Qp0zUprintf, Qp0zDump, Qp0zDumpStack, and Qp0zDumpTargetStack APIs. Refer to Application Program Interfaces (APIs) topic in the Information Center for more information on the user trace facility APIs.

The trace records written to the user trace buffer with the user trace facility APIs can be formatted and placed into a file or written to the *stdout* file by using the DMPUSRTRC (Dump User Trace Buffer) CL command.

User trace buffer spaces can be deleted by using the DLTUSRTRC (Delete User Trace Buffer) CL command.

Optional Parameters

JOB Specifies the job for which the user trace buffer is being changed.

*****: The user trace buffer for the job that the command is running in is changed.

job-name: Specify the name of the job whose user trace buffer is being changed. If no user name or job number qualifier is given, all of the jobs currently in the system are searched for the simple job name. If duplicates of the specified name are found, a qualified job name must be specified.

user-name: Specify the name of the user of the job whose user trace buffer is being changed.

job-number: Specify the six-digit number of the job whose user trace buffer is being changed.

CLEAR

Specifies whether all trace records currently stored in the user trace buffer space should be removed.

***NO**: No trace records are removed from the user trace buffer.

***YES**: All trace records currently stored in the user trace buffer are removed.

MAXSTG

Specifies the size, in kilobytes, that the user trace buffer will be created to (if it doesn't exist) or resized to (if it exists). If this parameter is specified, ***YES** must also be specified for the CLEAR parameter.

***SAME**: The user trace size is not changed. The default size (300 kilobytes) is used to create the user trace buffer when the first user trace API is called.

maximum-kilobytes: Specify the maximum amount of storage, in kilobytes, used to store user trace records. One kilobyte equals 1024 bytes.

TRCFULL

Specifies whether the trace records wrap (replace oldest records with new records) or whether the trace stops when all of the storage specified by the MAXSTG parameter has been used.

***SAME**: The current attribute does not change. The default when a user trace buffer space is created is TRCFULL(*WRAP).

***WRAP**: When the trace file is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

***STOPTRC:** Tracing stops when the trace buffer space is full of trace records.

Examples for CHGUSRTRC

Example 1: Changing the User Trace Buffer Size for the Current Job

```
CHGUSRTRC JOB(*) MAXSTG(100) CLEAR(*YES)
```

This command changes the user trace buffer size to 100 kilobytes.

Example 2: Clearing the User Trace Buffer for a Specific Job

```
CHGUSRTRC JOB(123581/DEPT2/WS1) CLEAR(*YES)
```

This command clears the user trace buffer for job WS1, which is associated with the user profile DEPT2, and has the job number 123581.

Error messages for CHGUSRTRC

*ESCAPE Messages

CPFA98A

A User Trace option could not be changed for job &3/&2/&1.

CPFA98C

Job &3/&2/&1 not unique.

CPF1070

Job &3/&2/&1 not found.

CHGVAR (Change Variable) Command Description

CHGVAR Command syntax diagram

Purpose

The Change Variable (CHGVAR) command is used in CL programs to change the value of a CL variable or to change part of a character variable by using the substring built-in function (%SUBSTRING) or the binary built-in function (%BINARY). The value can be changed to the value of a constant, to the value of another variable, or to the value gotten from the evaluation of an expression or a built-in function. Expressions and built-in functions are described in Expressions in CL Commands in the Information Center. Also, implicit conversion between decimal and character values is performed by the rules given in the VALUE parameter description.

The binary built-in function (%BINARY or %BIN) can be used in either the VAR or the VALUE parameter as a substitute for a decimal variable. When used with the VAR parameter, the specified portion of the character variable is changed to the signed binary integer equivalent value of the arithmetic expression given in the VALUE parameter. When used within the VALUE parameter, the specified portion of the character variable is treated as a signed binary integer converted to a decimal number when used in evaluating the value of the VALUE parameter. A 2-byte binary integer is converted to a decimal (5 0) number and a 4-byte binary number is converted to decimal (10 0) number. The result of the evaluated expression is then assigned to the specified in the VAR parameter.

The substring built-in function (%SUBSTRING or %SST) can be used in either the VAR or the VALUE parameter as a substitute for a character variable. When used with the VAR parameter, the specified portion of the character variable is changed to the value of the expression given in the VALUE parameter. When used within the VALUE parameter, the specified portion of the character variable is used in evaluating the value of the VALUE parameter. 2-byte binary integers are converted to a decimal

(50)numbers and a 4-byte binary numbers are converted to decimal (10 0) numbers. The result of the evaluated expression is then assigned to the specified in the VAR parameter.

The substring built-in function can be used to retrieve or change all or part of the local data area associated with a job.

The %SWITCH built-in function can be used in the VALUE parameter as a substitute for a logical variable declared in the program. %SWITCH contains an 8-character mask that indicates which of the eight job switches in a job are tested for 1s and 0s. When %SWITCH is specified for the VALUE parameter, the logical variable specified by the VAR parameter is set to '1' if the logical results of the built-in function are all true. If any of the job switches tested results in a false condition, the variable is set to '0'.

Restriction: The CHGVAR command is valid only in CL programs.

Required Parameters

VAR Specifies the name of the CL variable whose value is being changed. The type of variable does not have to be the same as the type of constant or variable specified in the VALUE parameter, unless an expression is being evaluated or the VAR parameter specifies a logical variable.

If the substring built-in function or the binary built-in function is used to change a portion of a character variable (that is, a substring of the character string in the variable) specified in VAR to a value specified in the VALUE parameter, specify the name of the character variable, followed by the starting position and the number of characters being changed within the character string specified by the variable name.

VALUE

Specifies the expression that is used to change the value of the variable. Variables, constants, or a built-in function can be used within the expression. For a description of expressions, see Expressions in CL Commands in the Information Center.

If a constant is used as a simple expression, its value must be specified by the following rules, depending on the type of constant being specified and whether the variable was declared as a decimal, character, or logical variable.

Coding Decimal Values for Decimal Variables

When a numeric value is specified for a decimal variable:

- It can be coded with or without a decimal point (. or ,) and with or without a plus or minus sign.
- If a negative value is specified, a minus sign (-) must precede the value.
- If a decimal point is not specified in the coded value, it is assumed to be on the right of the last digit specified; that is, the coded value is assumed to be an integer (whole number).
- If the number of either integer or fractional digits specified is greater than the defined number of integer or fractional digits, an error message is sent to the user.

For example, if a decimal variable is defined as a five-position decimal value of which two positions are the fraction portion, the following values can be coded:

Specified Value	Assumed Value
2.7 or 2,7	2.70
27 or 27.00	27.00
-27	-27.00

Coding Character Values for Decimal Variables

When a character value is specified for a decimal variable:

- Only the digits 0 through 9, a decimal point (. or), and a plus sign (+) or minus sign (-) can be used.
- If a plus sign or minus sign is specified, it must be placed immediately in front of (no blanks between) the first digit in the character value. If no sign character is specified, the value is converted as a positive value.
- The number of decimal positions in the converted result is determined by the decimal point specified in the character value. If no decimal point is specified, it is assumed to be to the right of the last digit in the converted value.
- Decimal alignment occurs in the converted result. The number of decimal positions in the converted result is determined by the number declared for the variable. If the specified character value has more decimal positions than the declared variable, the extra positions on the right are truncated. If the integer portion of the character value has more digits than that declared for the variable, an error message is sent to the user.

The following examples show the results of converting the indicated character values for character variable &A to decimal values for decimal variable &B.

```
CHGVAR VAR(&B) VALUE(&A)
```

Character Variable &A		Decimal Variable &B	
Length	Specified Value	Length	Converted Result
10	'+123.1'	5, 2	123.10
10	'+123.00'	5, 0	123
10	'-123'	5, 2	-123.00

When the binary built-in function is used instead of the decimal variable &B, the decimal value is converted to a signed binary number.

Coding Character Values for Character Variables

When a character string is specified for a character variable, it must be enclosed in apostrophes if it contains special characters or consists entirely of numeric characters. For example, 'ABC 67', which contains a blank, or '37.92', which contains a decimal point and consists entirely of numeric characters. If 37.92 is not enclosed in apostrophes, it is handled as a decimal value instead of a character value.

Character variables are padded with blanks (or are truncated) on the right if the character string for the VALUE parameter is shorter (or longer) than the variable specified by the VAR parameter.

If a character variable is set equal to a portion of another character variable, specify, as parameters on the substring built-in function, the name of the variable containing the substring, the starting character position, and the number of characters being replaced. The starting position and the number of characters can be specified in CL variables.

Coding Decimal Values for Character Variables

When a decimal value is specified for a character variable:

- The same digits, decimal point, and sign character (if the value is negative) are used in the converted result. The value is right-justified in the character variable and padded on the left with zeros, if needed (this is unique to converted CL decimal values).
- The converted result has as many decimal positions as were specified in the decimal value or as defined for the decimal variable being used. If no decimal positions are specified in the decimal value or defined for the decimal variable, no decimal point is placed in the result.
- A minus sign is placed in the leftmost position of the character variable if the specified decimal value is negative. No plus sign is placed in the character variable for positive values.

The following examples show the results of converting the indicated decimal values for decimal variable &B to character values for character variable &A.

```
CHGVAR VAR(&A) VALUE(&B)
```

When the binary built-in function is used instead of the decimal variable &B, the signed binary number is converted to a decimal number.

Decimal Variable &B		Specified Value	Character Variable &A	
Length			Length	Converted Result
5, 2		23.00 or +23	7	0023.00
5, 2		-3.9	7	-003.90
5, 2		-123.67	7	-123.67

Note:

The character variable must be long enough to accommodate the decimal point and sign character if the value can have a decimal point and a negative value in it. In the last example, although the decimal value is defined as (5, 2), the character variable must be at least 7 characters long for the value shown. In the next-to-last example, the character variable could only be 5 characters long and the converted result -3.90 would be valid.

The substring built-in function can be used to change a substring of a character variable specified in the VAR parameter to a decimal value in the VALUE parameter.

Coding Logical or Character Values for Logical Variables. The value for a logical variable must be a logical value of either '1' or '0'. It must be enclosed in apostrophes. However, the %SWITCH built-in function can be used in place of a logical variable in the VALUE parameter. Refer to for a description of the %SWITCH built-in function.

Note:

Values for decimal and character variable types can be specified in hexadecimal form (X'580F' for decimal 58.0). However, if character values are specified in hexadecimal form, care should be used because no validity checking is performed on the hexadecimal string.

The following examples of the CHGVAR command show how the values of decimal, logical, and character variables can be changed.

Examples for CHGVAR

Example 1: Changing Decimal Variables

```
CHGVAR &A &B
```

The value of variable &A is set to the value of the variable &B. If &B has a value of 37.2, then the value of &A becomes 37.2 also.

```
CHGVAR &Y (&Y#43; 1)
```

The value of variable &Y is increased by 1. If &Y has a value of 216, its value is changed to 217.

Example 2: Changing Logical Variables

```
CHGVAR &X (&Y *OR &Z)
```

The value of the logical variable &X is set to the value of the result of the OR operation of the logical variable &Y with the logical variable &Z. Both variables *must* be logical variables when *OR is used. If &Y equals '0' and &Z equals '1', then &X is set to '1'.

```
CHGVAR &A %SWITCH(10XXXX10)
```

The value of the logical variable &A is determined by the logical results of the built-in function, %SWITCH. Positions 1, 2, 7, and 8 of the 8-character mask indicate that the corresponding job switches for the job are to be tested for the values indicated in the mask. Job switches 1 and 7 are tested for 1s, and switches 2 and 8 are tested for 0s. (Switches 3 through 6 are not tested.) If all four switches contain the values specified in the %SWITCH mask, the logical result of the built-in function is true, and the variable &A is set to a '1'. If any of the four switches contain a value not indicated in the mask, the result is false and &A is set to '0'.

Example 3: Changing Character Variables

```
CHGVAR VAR(&A) VALUE(AB *CAT CD)
CHGVAR &A ('AB' *CAT 'CD')
```

These two commands set the value of the variable &A equal to the character string ABCD, which is the result of the concatenation of the two character strings AB and CD. The first command is coded in keyword form with unquoted strings; the second command is coded in positional form with the VALUE parameter specifying two quoted character strings.

```
CHGVAR &VAR1 &VAR2
```

This example shows a 6-character variable whose value is changed by a shorter character string. If &VAR1 = ABCDEF and &VAR2 = XYZ before the command is processed, the result in &VAR1 is padded on the right with blanks: XYZ .

```
CHGVAR &VAR1 '12'
```

Assuming &VAR1 is a character variable that is 6 characters long, the result is again padded on the right with blanks: 12 . The apostrophes are required in this example.

```
CHGVAR VAR(%SUBSTRING(&A 4 3)) VALUE(REP)
or
CHGVAR VAR(%SST(&A 4 3)) VALUE(REP)
```

The substring built-in function is used to change 3 characters of the character constant in the variable named &A. If &A has a value of ABCDEFGH, the fourth, fifth, and sixth characters in &A are set to REP, and the result is ABCREPGH.

```
CHGVAR VAR(%SST(*LDA 1 512)) VALUE(' ')
```

The substring built-in function is used to change all of the local data area to blanks.

```
CHGVAR VAR(%BINARY(&A 1 2)) VALUE(20)
```

or

```
CHGVAR VAR(%BIN(&A 1 2)) VALUE(20)
```

The binary built-in function is used to change the first 2 characters of the character variable named &A to the signed binary value of the number 20, or hexadecimal number X'0014'. If the character variable named &A has a length of 10, characters 3 through 10 of variable &A are not changed.

Error messages for CHGVAR

*ESCAPE Messages

CPF0816

%SWITCH mask &1 not valid.

CHGWSE (Change Work Station Entry) Command Description

CHGWSE Command syntax diagram

Purpose

The Change Work Station Entry (CHGWSE) command changes one or more attributes of a work station entry in the specified subsystem description.

Notes:

1. When the JOBD parameter is specified, the work station entry will be changed; however, the value of this parameter is not changed for any work stations started through this entry that are active at the time.
2. If the value of the MAXACT parameter is reduced to a number less than the total number of work stations that are active through the work station entry, no additional work stations will be allowed to sign on. Active work stations will not be signed-off; but, other work stations will not be allowed to sign on until the number of active work stations is less than the value specified for the MAXACT parameter.

Restriction: The user of this command must have object operational and object management authorities for the subsystem description, and object operational authority for the job description.

Required Parameters

SBSD Specifies the qualified name of the subsystem description that contains the work station entry that is changed.

The name of the subsystem description 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.

subsystem-description-name: Specify the name of the subsystem description that contains the work station entry that is being changed.

WRKSTN

Specifies the device description name of the work station whose work station entry is being changed.

The WRKSTN parameter and the WRKSTNTYPE parameter are mutually exclusive.

work-station-name: Specify the name of the work station whose work station entry is being changed.

generic-work-station-name:* Specify the generic name of the work station. 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. See generic names for additional information.

WRKSTNTYPE

Specifies the type of work station whose work station entry is being changed. This work entry applies to all work stations of this type that do not have specific work entries for an individual work station.

The WRKSTN parameter and the WRKSTNTYPE parameter are mutually exclusive.

The following type codes are valid:

Type Code

Device

3179	3179 Display Station
3180	3180 Display Station
3196	3196 Display Station
3197	3197 Display Station
3277	3277 Display Station
3278	3278 Display Station
3279	3279 Display Station
3476	3476 Display Station
3477	3477 Display Station
3486	3486 Display Station
3487	3487 Display Station
5251	5251 Display Station
5291	5291 Display Station
5292	5292 Color Display Station
5555	5555 Display Station (on systems supporting DBCS (double-byte character set))

***ALL:** The work station entry for all valid work station types is added.

***NONASCII:** The work station entry for all valid work stations that use 5250 data streams is added.

***ASCII:** The work station entries for all work stations that use ASCII data streams are added.

***CONS.:** This value overrides a device type entry that specifies the same device type as the device being use as the console.

work-station-type: Specify the work station device type whose work station entry is being changed.

Optional Parameters

JOBID Specifies the qualified name of the job description used for jobs that are created and processed through this work station entry. If the job description does not exist when this work station entry is changed, a library qualifier must be specified because the job description name is retained in the subsystem description.

***SAME:** The value does not change.

***USRPRF:** The job description named in the user profile of the user that signs on at this work station (or at this type of work station) is used for jobs started through this entry.

***SBSD:** The job description that has the same name as the subsystem description, specified by the SBSDB parameter, is used for jobs created through this entry.

The name of the job description 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.

job-description-name: Specify the qualified name of the job description that is used for jobs created through this entry.

MAXACT

Specifies, for work stations that use this work station entry, the maximum number of work station jobs that can be active (or signed on) at the same time. More information on this parameter is in Commonly used parameters.

***SAME:** The value does not change.

***NOMAX:** There is no maximum number of jobs that can be active at the same time.

maximum-active-jobs: Specify the new maximum number of jobs that can be active at the same time through this entry.

AT Specifies when the work stations associated with this work entry are allocated. For more information on how work stations are allocated to subsystems, see the Start Subsystem (STRSBS) command description.

Note:

If two or more work station entries specify AT(*SIGNON) for the same work station, if they are in more than one subsystem description, and if the work station is varied on while more than one of the subsystems are active, it cannot be predicted to which subsystem the work station will be assigned.

***SAME:** The value does not change.

***SIGNON:** The work stations are allocated when the subsystem is started. A sign-on prompt is shown at each work station associated with this work entry. If a work station becomes allocated to a different subsystem, interactive jobs associated with the work station are allowed to enter this subsystem through the Transfer Job (TFRJOB) command.

***ENTER:** The work stations associated with this work entry are not allocated when the subsystem is started. However, the interactive jobs associated with the work stations are allowed to enter this subsystem through the TFRJOB command.

Examples for CHGWSE

Example 1: Changing an Entry at Signon

```
CHGWSE SBSDB(QGPL/BAKER) WRKSTN(A12) AT(*SIGNON)
```

This command changes the work station entry for work station A12 in subsystem BAKER found in the general purpose library. A job is created for work station A12 when the user's password is entered on the sign-on prompt and the Enter key is pressed.

Example 2: Changing an Entry

```
CHGWSE SBSD(QGPL/BAKER) WRKSTN(B28)
```

This command changes the job entry for work station B28 in subsystem BAKER found in the general purpose library. The JOBID, MAXACT and AT parameters default to the *SAME value.

Error messages for CHGWSE

*ESCAPE Messages

CPF1619

Subsystem description &1 in library &2 damaged.

CPF1691

Active subsystem description may or may not have changed.

CPF1697

Subsystem description &1 not changed.

CHGWTR (Change Writer) Command Description

CHGWTR Command syntax diagram

Purpose

The Change Writer (CHGWTR) command allows the user to change the form type, number of file separators, and output queue attributes of an active printer writer. This capability provides the best performance when you want to process all files of one form type or output queue, and then all files of another form type or output queue, instead of ending and restarting the writer or changing to a different output queue each time.

If changes are made while the writer is in hold (HLD) status, the changes do not take effect until after the writer is released. The change is then made based on the value specified on the OPTION parameter.

Required Parameter

WTR Specifies the simple name of the printer writer being changed. Each writer name must be unique.

***SYSVAL:** The writer name of the system default printer is changed.

writer-name: Specify the name of the writer being changed.

Optional Parameters

OUTQ Specifies the qualified name of the output queue.

***SAME:** The output queue being used is not changed.

***DEV:** Specifies the default output queue associated with the printer is used.

The name of the output 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, the QGPL library is used.

library-name: Specify the name of the library to be searched.

output-queue-name: Specify the name of the output queue.

FORMTYPE

Specifies the form type selection codes for which the spooled files are being produced by the writer. A file's form type is originally derived from the form type specified in the device file that produced the spooled file. This parameter specifies that only the files of this form type are produced by the writer. All other files are left on the output queue as available.

Note:

The form load message is issued when the spooled file to be printed has a form type different from the form type of the last spooled file that was printed on the device. The last form type printed is kept from the last STRPRTWTR, CHGWTR, or VRYCFG command issued.

Consider the following example:

1. The last spooled file printed on printer PRT01 had the form type *STD.
2. The user changes the form type on PRT01 to XYZ using the following command:
`CHGWTR PRT01 FORMTYPE(XYZ)`
3. No spooled file with the form type XYZ is printed on PRT01.
4. The user then sends a spooled file with the form type *STD to PRT01. The form load message is not issued, despite the intervening CHGWTR command, because the last spooled file printed on PRT01 had the same form type as the spooled file being printed.
The form load message would be issued if a spooled file with the form type XYZ were actually printed on PRT01.

Element 1: Type of Form Designation

***SAME:** The value does not change.

***FORMS:** All available files with the same form type are produced as a group before the writer moves on to the next form type. The writer initially chooses the first available file on the queue. After the first file is complete, all files with the same form type as the first are processed. The writer again chooses the first available file on the queue and repeats the process.

***ALL:** All form types are produced by the writer.

***STD:** Only files that specify the standard form type are selected.

form-type: Specify the type of form for which the spooled files are produced.

Element 2: Message Sending Options

***SAME:** The value does not change.

***INQMSG:** An inquiry message is sent when the current file has a form type that is different than the one loaded on the device.

***INFOMSG:** An informational message is sent when the writer runs out of files with the specified form type.

***NOMSG:** No informational message is sent when the writer runs out of files with the specified form type. Also, no inquiry message is sent when the current file has a form type that is different than the one loaded on the device.

***MSG:** The writer sends an informational message when it runs out of files with the specified form type. If the printer writer does not end, (specified by the AUTOEND parameter on the STRPRTWTR command), then as additional spooled files become available for printing they are produced. The informational message is sent each time the writer must wait for additional spooled files.

FILESEP

Specifies the number of file separator pages to print preceding each file.

***SAME:** The value does not change.

***FILE:** Print the number of file separator pages that is specified for each individual file.

number-of-file-separators: Specify the number of file separator pages to print.

SEPDRAWER

Specifies which drawer is selected for printing file and job separators.

***SAME:** The value does not change.

***DEVD:** The value stored in the device description for the printer is used.

***FILE:** The separator pages are printed from the same drawer as the spooled file.

separator-drawer: Specify a value ranging from 1 through 255 to indicate the drawer from which the separator pages are printed.

Note:

For some printers, SEPDRAWER(3) implies an envelope drawer.

OPTION

Specifies when the writer change occurs.

***NORDYF:** The writer change occurs when there are no files on the output queue that meet the writer's current form type selection requirements.

***FILEEND:** The writer change occurs at the end of the current file.

Example for CHGWTR

```
CHGWTR WTR(MYWTR) FORMTYPE(MYFORM *NOMSG)
OPTION(*FILEEND)
```

This command changes writer MYWTR, which has been producing files of some other form type, to produce files with a form type of MYFORM at the end of the file now being produced. The writer is also prevented from sending an informational message when it runs out of eligible files with form type MYFORM.

Additional Considerations

By using the CHGWTR command to control the form type being produced by a writer, the operator can improve efficiency by minimizing the number of form changes required. On the other hand, if the print writer's output queue has a large number of spooled files with the wrong form type, which must be bypassed by the writer, then writer output queue search time increases and total system performance may decrease.

It is possible to enter another change writer command for a writer before the previous change has taken place. When this is done the later change is not queued, but it updates the first change instead. For example, if the command

```
CHGWTR WTR(WRKL) OUTQ(QPRINT) OPTION(*FILEEND)
```

is entered, then (before the change takes place) the command

```
CHGWTR WTR(WRKL) FORMTYPE(XYZ *SAME)
  OPTION(*FILEEND)
```

is entered, the output queue of writer WRKL changes to QPRINT, and the form type is changed to XYZ when the writer finishes producing the current file.

Whenever the output queue or form type is changed for a writer, the print writer starts searching the output queue from the beginning and selects the first available file on the queue. This occurs even if the change takes place between two SCHEDULE(*JOBEND) files of the same job.

Error messages for CHGWTR

***ESCAPE Messages**

CPF1842

Cannot access system value &1.

CPF2207

Not authorized to use object &1 in library &3 type *&2.

CPF3313

Writer &1 not active nor on job queue.

CPF3330

Necessary resource not available.

CPF3331

Not authorized to control writer &3/&2/&1.

CPF3357

Output queue &1 in library &2 not found.

CPF3456

Cannot change writer &1 to output queue &4 in library &5.

CPF3457

Cannot change writer &1.

CPF3458

Change writer &1 not allowed. End writer pending.

CPF3459

Writer &1 not eligible for change.

CPF3460

Change writer &1 not allowed.

CPF3463

Output queue for device &1 not found.

CPF3464

Not authorized to output queue &1 in library &2.

CPF9803

Cannot allocate object &2 in library &3.

CHKASPBAL (Check ASP Balance) Command Description

CHKASPBAL Command syntax diagram

Purpose

The Check ASP Balance (CHKASPBAL) command allows the user to check which auxiliary storage pool (ASP) balance function is currently active and which units have been marked to not allow new allocations (*ENDALC). Informational messages will be sent to the job log indicating which ASP function is active and which units are marked *ENDALC. Message CPIB715 will indicate which ASP balancing function is active. Message CPIB716 will indicate no ASP balancing is active. Message CPIB714 will indicate that no units are marked *ENDALC. Message CPIB713 is issued for each unit marked *ENDALC.

This command has no parameters.

Example for CHKASPBAL

Example 1: Checking ASP Balancing

CHKASPBAL

This command will check which ASP balance function is currently active and which units, if any, are marked to not allow new allocations (*ENDALC). Informational messages will be sent to the job log for each configured ASP indicating which balance operation is active. An informational message will be sent to the job log for each unit that is marked *ENDALC.

Error messages for CHKASPBAL

None <<

CHKCMNTRC (Check Communications Trace) Command Description

CHKCMNTRC Command syntax diagram

Purpose

The Check Communications Trace (CHKCMNTRC) command returns the communications trace status for a specific line, network interface description, network server description, or for all of the traces of a specific type that exist on the system. The status is returned through a message.

Restrictions:

1. To use this command you must have *SERVICE special authority, or be authorized to the Service Trace function of Operating System/400 through iSeries Navigator's Application Administration support. The Change Function Usage Information (QSYCHFUI) API, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
2. The following user profiles have authority to this command:
 - QSECOFR
 - QSRV

Required Parameters

CFGOBJ

Specifies the name of the configuration description to check. The configuration description is either a line description or a network interface description.

***ALL:** The communications trace status is returned for all of the traces of a specific type.
configuration-name: Specify the configuration object name.

CFGTYPE

Specifies the object type of the configuration description to check.

***LIN:** Status for lines is shown.

***NWI:** Status for network interfaces is shown.

***NWS:** Status for network servers is shown.

Examples for CHKCMNTRC

Example 1: Checking All Traces

```
CHKCMNTRC CFGOBJ(*ALL) CFGTYPE(*NWI)
```

This command shows the communications trace status of all network interface traces.

Example 2: Checking An Individual Trace

```
CHKCMNTRC CFGOBJ(*QESLINE) CFGTYPE(*LIN)
```

This command shows the communications trace status of line QESLINE.

Error messages for CHKCMNTRC

***ESCAPE Messages**

CPF39A7

Trace storage not available in communications processor

CPF39A8

Not authorized to communications trace service tool

CPF39A9

Error occurred during communications trace function

CPF39BE

No communications traces of type &1 exist

CPF39B0

No communications traces exist.

CPF39B1

Trace &1 type &2 does not exist

CPF39B6

Communications trace function cannot be performed

CHKIGCTBL (Check DBCS Font Table) Command Description

CHKIGCTBL Command syntax diagram

Purpose

The Check DBCS Font Table (CHKIGCTBL) command checks for the existence of a specified double-byte character set (DBCS) font table. Use this command to verify that one of the DBCS font tables in the system prints and shows characters in the matrix pattern used by a given device. If the table does not exist, the system sends you a message. If the table exists, the system does not send you a message.

DBCS font tables contain the images in a given dot matrix of the DBCS extension characters used on the system. The system refers to the tables when printing and showing these characters. There are separate tables for each character image matrix used by devices attached to the system.

Required Parameter

IGCTBL

Specifies the name of the DBCS font table for which its existence is being checked. Choose one of the following table names:

QIGC2424: The Japanese DBCS font table used for showing and printing extension characters in a 24-by-24 dot matrix image.

QIGC2424C: The Traditional Chinese DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC2424K: The Korean DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC2424S: The Simplified Chinese DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC3232: The Japanese DBCS font table used for showing and printing extension characters in a 32-by-32 dot matrix image.

QIGC3232S: The Simplified Chinese DBCS font table used for printing extension characters in a 32-by-32 dot matrix image.

QIGCrrcccl: The name of the DBCS font table checked for must always be in the format QIGCrrcccl, where *rr* is the table row matrix size, *cc* is the table column matrix size, and *l* is an optional language identifier.

Example for CHKIGCTBL

```
CHKIGCTBL IGCTBL(QIGC2424)
```

This command causes the system to check for the Japanese DBCS font table that contains character images in a 24-by-24 dot matrix image.

Error messages for CHKIGCTBL

*ESCAPE Messages

CPF8421

DBCS font table &1 not found.

CHKDKT (Check Diskette) Command Description

CHKDKT Command syntax diagram

Purpose

The Check Diskette (CHKDKT) command searches a diskette in a specified device for a unique volume identifier and/or file label. If the volume identifier and/or file label are found on that diskette, a successful completion message is sent to the user who entered the command for that diskette. The diskette may be processed on the next diskette operation. If the correct diskette is not found, a message is sent to the user who entered the command. A check for this message in a CL program can direct the logic flow depending on whether the correct diskette is in the drive.

The diskette specified by the DEV parameter is searched. If a volume identifier is specified in the command, the volume identifier of the diskette is compared with the volume identifier in the command. If

the correct volume identifier is found or if no volume identifier was specified in the command and a data file identifier is specified, the volume table of contents (VTOC) of the diskette is checked for the specified label.

If the correct data file identifier is found and the creation date is specified on the command, the date in the data file identifier is compared with that of the command. If they match, the correct file is found. If they do not match, the remaining labels in the VTOC are checked for both that data file identifier and the specified creation date.

If a match of the specified parameters is not found on the diskette, a message is sent to the user who entered the command.

Note:

The identifiers are checked on each diskette in the following order:

1. Volume identifier
2. Data file identifier
3. The date it was created

Each parameter is checked on the diskette if (and only if) the parameters before it in the list are found on that diskette or were not specified in the command. If a match of all parameters in the command is not found on the diskette, a message is sent to the user who entered the command.

Since this command can be used in a CL program to determine whether the diskette is to be processed, the message for a media error while reading the VTOC is not sent to the user. Instead, a status message is sent to the user who entered the command. The status message can be checked by giving control to the CL program. If the status message is not checked, a message is sent to the user who entered the command.

Restriction: If a diskette has an extended label area, that extension area is not checked when searching for a file label. An informational message is sent notifying the user of this omission.

Note:

Results when processing diskettes with labels that are not IBM standard labels are unpredictable. Initialize the diskette by specifying CHECK(*NO) on the Initialize Diskette (INZDKT) command.

Required Parameter

DEV Specifies the name of the device in which the diskette being checked is located.

Optional Parameters

VOL Specifies one or more volume identifiers used by the file. More information on this parameter is in Commonly used parameters.

***MOUNTED:** The volume currently placed in the device is used.

volume-identifier: Specify the volume identifier being compared with the volume identifier field on the diskette label. The identifier can have a maximum of 6 characters. Any combination of letters and numbers can be used. If the volume identifier does not match the diskette, an escape message is sent.

LABEL

Specifies whether a check is made for a specific file label on the diskette.

***NONE:** No data file identifier check is made.

data-file-identifier: Specify the data file identifier that is being checked for. A search for that data file identifier is done on the diskette in the specified device.

CRTDATE

Specifies whether the creation date of the data file identifier being checked is also checked.

***NONE**: The file creation date is not checked. If LABEL(*NONE) is specified, CRTDATE(*NONE) must also be specified.

creation-date: Specify the date that must match the creation date of the file being checked. The date must be entered in the format specified for the system values QDATFMT and, if separators are used, QDATSEP. When the correct file label is found, the creation date in that data file identifier is compared with the value in this parameter. If it does not match, the next data file identifier in the VTOC is checked.

Examples for CHKDKT

Example 1: Checking a Volume Identifier

```
CHKDKT DEV(QDKT) VOL(MASTER)
```

This command checks the volume identifier of the diskette in device QDKT for a volume identifier of MASTER. If the volume identifier of the diskette is MASTER, a completion message is sent. If the volume identifier of the diskette is not MASTER, a message is sent indicating that the volume identifier is incorrect and the job must be sent again.

Example 2: Checking Volume Identifier and File Creation Date

```
CHKDKT DEV(QDKT) VOL(VOLID) LABEL(FILE)  
CRTDATE('7/4/76')
```

This command searches the diskette in device QDKT for a volume identifier of VOLID. If a diskette is found with that volume identifier, the file labels on the diskette are checked for a data file identifier of FILE. If that data file identifier is found and the creation date of the file is 7/4/76, the correct file and diskette have been found, and a completion message is sent. If the correct volume identifier and data file identifier and creation date are not found, a message is sent to the user indicating that the volume identifier is incorrect and the job must be sent again.

Error messages for CHKDKT

***ESCAPE Messages**

CPF6162

Diskette does not contain specified identifiers.

CPF6708

Command ended due to error.

CPF6716

Device &1 not a diskette device.

CPF6718

Cannot allocate device &1.

CPF9814

Device &1 not found.

CPF9825

Not authorized to device &1.

CHKDLO (Check Document Library Object) Command Description

CHKDLO Command syntax diagram

Purpose

The Check Document Library Object (CHKDLO) command verifies that an object exists and that a user has authority to the object before trying to access it.

These checks are useful before the user tries to access several objects at the same time. The CHKDLO command is also used to check the validity of document library object names contained in CL variables and to verify object authorizations under program control.

When the command runs, the system searches for the specified object. If the object is found, the system verifies that the user is authorized to that object in the manner specified on the CHKDLO command. If the object is not found or the user does not have the authority specified on the CHKDLO command, an escape message is sent to the user.

When the CHKDLO command is used in a CL program, at least one Monitor Message (MONMSG) command should follow the CHKDLO command to monitor messages that result from running this command. More information on this parameter is in Commonly used parameters.

Required Parameter

DLO Specifies the document library object that is checked.

***SYSOBJNAM:** The system object name is used to identify the document or folder that is checked. This parameter must be used to check a document that is not in a folder. It may also be used instead of a folder name or document name, whenever the system object name is known. The SYSOBJNAM parameter and DLO(*SYSOBJNAM) must be specified together.

document-name: Specify the name of the document that is checked.

folder-name: Specify the name of the folder that is checked.

Optional Parameters

FLR Specifies the name of the folder that contains the document.

***NONE:** The object that is checked is not contained in a folder.

folder-name: Specify the name of the folder that contains the document or folder that is checked. A folder name can be specified only if a folder or document name is specified for the DLO parameter.

SYSOBJNAM

Specifies the system object name. This parameter is valid only when DLO(*SYSOBJNAM) or DOCL(*SYSOBJNAM) is specified. A full ten characters must be specified.

OBJTYPE

Specifies the OS/400 system object type of the document library object that is checked. OBJTYPE(*DOC) cannot be specified when a document or folder name is specified for the DLO parameter, and FLR(*NONE) is also specified. More information on this parameter is in Commonly used parameters.

***ANY:** The object that is checked is either a document or a folder.

***DOC:** The object that is checked is a document.

***FLR:** The object that is checked is a folder.

AUT Specifies the authority being checked. More information on this parameter is in Commonly used parameters.

***NONE:** Authority is not checked.

***ALL:** The user is checked for authority to perform all operations on the document or folder except those limited to the owner.

***CHANGE:** The user is checked for authority to perform all operations on the document or folder except those limited to the owner or 'ALL' authority.

***USE:** The user can perform basic operations on the document library object, such as running a program or reading a file. The user cannot change the document library object. *USE authority provides object operational authority, read authority, and execute authority.

***EXCLUDE:** The user cannot access the document library object.

USRID

Specifies the user ID and address of the user for whom the object is being checked. If a user ID and address of someone other than the user who is signed on is specified,

the user must have *ALLOBJ special authority or both users must be enrolled in the system directory and the user who is signed on must be granted permission (using the GRTUSRPMN command) to work on behalf of the specified user.

***CURRENT:** The user profile that is currently running is used.

Element 1: User ID

user-ID: Specify the user ID of the user on whose behalf the object is to be checked.

Element 2: User Address

user-address: Specify the address of the user on whose behalf the object is to be checked.

Note:

The USRID parameter is useful only when the AUT parameter and any special value are specified, except *NONE.

Example for CHKDLO

```
CHKDLO DLO(FLR1) OBJTYPE(*ANY) AUT(*NONE)
      USERID(USER1 ADDR1)
```

This command checks for the existence of a folder named FLR1 on behalf of a user whose user ID is USER1 and whose address is ADDR1. The user's authority to FLR1 is not checked.

Error messages for CHKDLO

*ESCAPE Messages

CPF8A11

CHKDLO command failed.

CPF8A75

Not authorized to access folder &1.

CPF8A77

Folder &1 not found.

CPF8A82

Document &2 not found in folder &1.

CPF8A83

Not authorized to access document &2 in folder &1.

CHKEXPBRM (Check Expired Media for BRM) Command Description

Note: To use this command, you must have the 5722-BR1 (Backup Recovery and Media Services for iSeries) licensed program installed. For detailed information on the parameters of this command, see the online help.

CHKEXPBRM Command syntax diagram

Purpose

The Check Expired Media for BRM (CHKEXPBRM) command is used to check to see if there is enough expired media to satisfy the media requirements of a save operation. The command calculates the media of a specified media class available for a save operation, taking into account the location of the media based on the location specified in the media policy or a location that you specify. You can calculate the available media for single or multiple media classes, with single or multiple location combinations depending on the values that you specify in the command.

For instance, if you specified 20 in the Required volumes field, FMT2GB in the Media class field and *HOME in the Location field, you would have one media class, location combination. If you changed the Location field to *ANY, you could potentially receive messages about media class availability for each of the locations that you have set up that have the specified media class. If you had a situation where you specified *MEDPCY in the Required volumes field, *CTLGRP in the Media policy field and *ALL in the Control group field, then you would expect to have multiple media class, location combinations. The command returns messages that specify whether enough media is available for each media class, location combination.

This command works in conjunction with the value specified in the **Required volumes** field. The value can be a specified number of volumes or a special value for media policy. The number of available volumes calculated by the CHKEXPBRM command is compared against the value in the **Required volumes** field. If the expired media calculated by the CHKEXPBRM command is greater than or equal to the **Required volumes** field value, the save operation can continue. If the value is less, then the save operation is canceled.

The number of expired volumes can also be checked by user jobs using the CHKEXPBRM command. For example, the CHKEXPBRM command could be incorporated into a job scheduler and used to determine at various times if there is enough expired media available for a save operation.

This command is used by all BRMS save operations.

Example for CHKEXPBRM

Example 1: Checking on Available Volumes

```
CHKEXPBRM EXPMED(50) MEDCLS(FMT2GB) LOC(*HOME)
```

In this example, you are checking to see if there are 50 expired volumes of media class FMT2GB available for a save operation at the home location.

Error messages for CHKEXPBRM

None

CHKIN (Check In) Command Description

CHKIN Command syntax diagram

Purpose

The Check In (CHKIN) command checks in an object that had previously been checked out.

For more information about integrated file system commands, see the Integrated file system topic in the File systems and management category of the Information Center.

Restrictions:

1. To check in an object that someone else has checked out, the user must own the object or have one of the following:
 - *ALL authority to the object
 - *ALLOBJ special authority
2. The user must have execute authority to each directory in the path.
3. Not all file systems support the CHKIN command.

Required Parameter

OBJ Specifies the path name of the object or a pattern to match the path name or names of objects to be checked in.

The object path name can be either a simple name or a name that is qualified with the name of the directory in which the object is located. A pattern can be specified in the last part of the path name. An asterisk (*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes.

Example for CHKIN

```
CHKIN OBJ('W')
```

This command checks in file W in your current directory. If another user has the file checked out and this user has sufficient authority, the file is checked in.

Error messages for CHKIN

*ESCAPE Messages

CPFA09C

Not authorized to object.

CPFA09D

Error occurred in program &1.

CPFA0A1

An input or output error occurred.

CPFA0A3

Path name resolution causes looping.

CPFA0A7

Path name too long.

CPFA0A9

Object not found.

CPFA0AB

Object name not a directory.

CPFA0AD

Function not supported by file system.

CPFA0B2

No objects satisfy request.

CPFA0BE

&1 objects checked in. &2 objects failed.

CPFA0BF

&1 objects checked out. &2 objects failed.

CPFA0DA

Object name is a directory.

CHKOBJ (Check Object) Command Description

CHKOBJ Command syntax diagram

Purpose

The Check Object (CHKOBJ) command verifies that an object exists and that a user has authority to the object before access to it is permitted. For verification, as many as ten specific authorities can be specified in the command.

These checks are particularly useful before the user tries to access several objects at the same time. The CHKOBJ command is also used to check the validity of object names contained in CL variables and to verify object authorizations under program control.

When the command runs, the system searches for the specified object. If the object is found, the system verifies that the user is authorized to that object as specified on the CHKOBJ command. If the object is not found or the user does not have the authorities specified on the CHKOBJ command, an escape message is sent to the user.

When the CHKOBJ command is used in a CL program, at least one Monitor Message (MONMSG) command should follow the CHKOBJ command to monitor for any messages that result from running this command.

Required Parameters

OBJ Specifies the qualified name of the object being checked. If no library name is given, *LIBL is used to find the object.

The name of the object 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.

object-name: Specify the name of the object that is checked.

OBJTYPE

Specifies the object type of the OS/400 system object being checked. Enter the predefined value that specifies the object type. More information on this parameter is in Commonly used parameters.

Optional Parameters

MBR Specifies which logical file member is checked, when checking data authorities.

Note: The logical file member, and the physical file members on which it is based are checked.

***NONE:** Database file members are not checked, but the existence and (optionally) the authority of the file are checked. For all other object types (including device files), *NONE is the only valid value for the MBR parameter.

***FIRST:** The first member in the database file is used.

database-file-member-name: Specify the name of a physical or logical file member to be checked by the CHKOBJ command. Values specified for the OBJ and OBJTYPE parameters must identify a database file and the member specified must be a member of the database file specified in the OBJ parameter.

AUT Specifies the authority being checked.

***NONE:** The user's authority is not checked.

***USE:** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. *USE authority provides object operational authority, read authority, and execute authority.

***CHANGE:** The user's object operational authority and all data authorities to the object are checked regardless of the object type.

***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 also can change ownership of the object.

***EXCLUDE:** The user cannot access the object.

***AUTLMGT:** The user's authority to add, delete, or change users and their authorities on the authorization list or delete the authorization list is checked.

Note: The OBJTYPE(*AUTL) parameter must be specified before specifying AUT(*AUTLMGT).

***OBJALTER:** Object alter authority provides the authority needed to alter the attributes of an object. If the user has this authority on a database file, the user can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. If the user has this authority on an SQL package, the user can change the attributes of the SQL package. This authority is currently only used for database files and SQL packages.

***OBJEXIST:** The user's authority to control object ownership and existence is checked. These authorities are required for a user to delete an object; to free storage; or to save, restore, or transfer ownership of the object. A user with special save system (*SAVSYS) authority does not need object existence authority to save or restore the object.

***OBJMGT:** The user's authority to manage the access to, and availability of, the object is checked. A user with object management authority can check personal authority status, move and rename objects, and add members to database files.

***OBJOPR:** The user's authority to manage access to, and availability of, the object is checked. Object operational authority has no data authority associated with it. Data authorities (listed below) are individually checked.

***OBJREF:** Object reference authority provides the authority needed to reference an object from another object such that operations on that object may be restricted by the other object. If the user has this authority on a physical file, the user can add a referential constraint in which the physical file is the parent. This authority is currently only used for database files.

***ADD:** The user's add authority, which is needed to add entries to the object (for example, adding job entries to a queue or adding records to a file) is checked.

***DLT:** Delete authority allows the user to remove entries from an object, for example, remove messages from a message queue or records from a file.

***EXECUTE:** The user's execute authority, which is needed to run a program or locate an object in a library.

***READ:** The user's read authority, which is needed to retrieve the contents of the object entry is checked.

***UPD:** The user's update authority, which is needed to update entries in the object is checked.

Examples for CHKOBJ

Example 1: Checking for Existence of a Program

```
CHKOBJ OBJ(LIB1/PROG1) OBJTYPE(*PGM)
```

This command checks for the existence of a program named PROG1 in library LIB1. The user's authorities to PROG1 are not checked.

Example 2: Checking for User's Authority to File

```
CHKOBJ OBJ(SOURCE1) OBJTYPE(*FILE)  
      MBR(MBR3) AUT(*CHANGE)
```

This command checks for the existence of file SOURCE1 and for the existence of member MBR3 in file SOURCE1. It also checks to see if the user has *CHANGE authority to file SOURCE1.

Example 3: Checking for User's Authority to Program

```
CHKOBJ OBJ(LIB1/PROG1) OBJTYPE(*PGM) AUT(*CHANGE)
```

This command checks the existence of program PROG1 in library LIB1. It also checks to see if the user has *CHANGE authority to PROG1.

Messages that can be monitored by the Monitor Message (MONMSG) command if sent by the CHKOBJ command are:

CPF9801

OBJECT NOT FOUND-PROG1 does not exist.

CPF9802

OBJECT NOT AUTHORIZED-The user that issued this command does not have *CHANGE authority to PROG1.

CPF9810

LIBRARY NOT FOUND-LIB1 cannot be located.

CPF9820

NOT AUTHORIZED TO LIBRARY-The user that issued this command is not authorized to the library named LIB1.

CPF9830

UNABLE TO ALLOCATE LIBRARY-The library named LIB1 is locked and cannot be accessed.

Example 4: Checking User's Authority to a Logical File Member

```
CHKOBJ OBJ(FILEA) OBJTYPE(*FILE)
      MBR(MBR1) AUT(*USE)
```

This command checks the user's authority to use logical file member MBR1, and each physical file member on which MBR1 is based.

In addition to the messages listed in the previous example, messages that can be monitored by the MONMSG command if sent by the CHKOBJ command, are:

CPF9815

MEMBER IN FILE NOT FOUND-MBR1 cannot be found in FILEA or FILEA does not contain members. If FILEA is a device file, a CPF9899 message is sent.

CPF9899

FUNCTION NOT PERFORMED-This message is a summary escape message that is always preceded by a diagnostic message. If FILEA is a device file, message CPD2168 precedes message CPF9899. If FILEA is locked, message CPF3202 precedes this message.

Example 5: Checking User's Add and Delete Authority

```
CHKOBJ OBJ(FILEA) OBJTYPE(*FILE) MBR(MBR1)
      AUT(*ADD *DLT)
```

```
MONMSG MSGID(CPF9802) EXEC(GOTO ERROR1)
```

These two commands (CHKOBJ and MONMSG) are used to verify that the user has both add and delete authority for logical file FILEA and each of the physical file members on which the logical file member MBR1 in the logical file FILEA is based. If the user does not have data authority for FILEA and each of the physical file members on which FILEA is based, escape message CPF9802 is sent to the program, and control in the program is passed to the command that has the label ERROR1.

Error messages for CHKOBJ***ESCAPE Messages****CPF9801**

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9810

Library &1 not found.

CPF9815

Member &5 file &2 in library &3 not found.

CPF9820

Not authorized to use library &1.

CPF9830

Cannot assign library &1.

CPF9899

Error occurred during processing of command.

CHKOBJITG (Check Object Integrity) Command Description

CHKOBJITG Command syntax diagram

Purpose

The Check Object Integrity (CHKOBJITG) command checks the objects owned by the specified user profile, the objects that match the specified path name, or all objects on the system to determine if any objects have integrity violations. An integrity violation occurs if:

- a command has been tampered with.
- an object has a digital signature that is not valid.
- the object has an incorrect domain attribute for the object type.
- a program or module object has been tampered with.
- >> a library's attributes have been tampered with.<<

If an integrity violation has occurred, the object name, library name (or pathname), object type, object owner, and type of failure are logged to a database file.

Also logged to the database file, but not an integrity violation are objects that do not have a digital signature but can be signed, objects that could not be checked, and objects whose format requires changes to be used on this machine implementation (IMPI to RISC conversion).

Note: Objects that are compressed, damaged, saved with storage freed, or in debug mode may not be checked. >>

Note: IBM commands duplicated from a release prior to V5R2 will be logged as ALTERED violations. These commands should be deleted and re-created using the CRTDUPOBJ (Create Duplicate Object) command each time a new release is loaded. <<

Restriction: To check object integrity, you must have *AUDIT special authority.

Note:

The CHKOBJITG command may run a long time if:

- the user profile specified for the USRPRF parameter owns many objects.
- many objects match the path name pattern specified for the OBJ parameter.
- *ALL is specified for the USRPRF parameter.
- *SYSTEM is specified for the OBJ parameter.

Required Parameters

USRPRF

Specifies the user profile or generic user profile name for which owned objects are checked for integrity violations.

Note: A value must be specified for either the USRPRF parameter or the OBJ parameter. You cannot specify values for both parameters.

***ALL:** Objects owned by all user profiles on the system are to be checked.

generic-user-name:* Specify the generic user profile whose owned objects are to be checked.

user-name: Specify the user profile whose owned objects are to be checked.

OBJ Specifies the objects that will be checked for integrity violations.

Note: A value must be specified for either the USRPRF parameter or the OBJ parameter. You cannot specify values for both parameters.

***SYSTEM:** All objects  in all available auxiliary storage pools (ASPs)  are to be checked.

Note: When *SYSTEM is specified, the only value allowed for the CHKSIG parameter is *ALL.

path-name: Specify the path name of the objects that are to be checked.

OUTFILE

Specifies the qualified name of the database file to which the output is directed. If the file does not exist, this command creates a database file in the specified library. If this function creates the file, the text reads *Outfile* for CHKOBJITG and the public authority is *EXCLUDE.

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.

database-file-name: Specify the name of the database file.

OUTMBR

Specifies the name of the database file member to which the output is directed.

Element 1: Member to Receive Output

***FIRST:** The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified on the OUTFILE parameter. If the member exists, the user can choose to either add records to the end of the existing member or to clear the existing records in the member and then add the new records.

member-name: Specify the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

Element 2: Operation to Perform on Member

The system clears the existing member and adds the new records.

The system adds the new records to the end of the existing records.

CHKDMN

Check object domain integrity.

***YES:** Object domain integrity is to be checked.

Note:

The following objects are valid in user domain so they are not checked:

- QTEMP library
- all objects of type *PGM
- all objects of type *SQLPKG
- all objects of type *SRVPGM

The following object types are valid in user domain only if the library they are in is specified in system value QALWUSRDMN (or if QALUSRDMN is *ALL).

- *USRSPC
- *USRQ
- *USRIDX

***NO:** Object domain integrity is not to be checked.

CHKCMD

The integrity of commands are checked.

***YES:** Command integrity is to be checked.

***NO:** Command integrity is not to be checked.

CHKPGMMOD

The integrity of program and module objects are checked.

***YES:** Program and module integrity is to be checked.

***NO:** Program and module integrity is not to be checked.

CHKSIG

The digital signatures of objects are to be checked.

***SIGNED:** Objects with digital signatures are checked. Any object with a signature that is not valid will be logged.

***ALL:** All objects that can be digitally signed are checked. Any object that can be signed but has no signature will be logged. Any object with a signature that is not valid will be logged.

***NONE:** Digital signatures will not be checked.

SUBTREE

Check the subtrees of the directories. (This parameter is only used when the OBJ parameter is specified).

***ALL:** All directory subtrees are checked.

***NONE:** No directory subtrees are checked.

» CHKLIB

The integrity of library attributes is checked.

***YES:** Library attribute integrity is to be checked.

***NO:** Library attribute integrity is not to be checked. <<

Examples for CHKOBJITG

Example 1: Check Objects Owned by One User Profile

```
CHKOBJITG USRPRF(JOEPGMR) OUTFILE(SECCHECK)
  OUTMBR(*FIRST *REPLACE) CHKDMN(*YES)
  CHKPGMMOD(*YES) CHKSIG(*YES) CHKLIB(*YES)
```

This command checks all objects owned by user JOEPGMR for integrity violations. Objects with an incorrect domain, program and module objects that have been tampered with, objects with digital signatures that are not valid, >> and libraries whose attributes have been tampered with << will cause integrity violation records to be logged in database file SECCHECK. Database file SECCHECK is first cleared of any existing records.

Example 2: Check Objects Owned by Multiple User Profiles

```
CHKOBJITG USRPRF(ABC*)
  OUTFILE(ABCCHECK) OUTMBR(*FIRST *REPLACE)
  CHKDMN(*YES) CHKPGMMOD(*YES) CHKSIG(*NONE) CHKLIB(*YES)
```

This command checks all objects owned by user profiles that start with ABC for integrity violations. Objects with an incorrect domain, program and module objects that have been tampered with, >> and libraries whose attributes have been tampered with << will cause integrity violation records to be logged to database file ABCCHECK. Database file ABCCHECK will first be cleared of any existing records.

Example 3: Check Objects in One Library

```
CHKOBJITG OBJ('/QSYS.LIB/LIB2.LIB/ABC*.*')
  OUTFILE(SECCHECK2) OUTMBR(*FIRST *REPLACE)
  CHKDMN(*YES) CHKPGMMOD(*YES) CHKSIG(*ALL) CHKLIB(*NO)
```

This command checks objects in library LIB2 that have names beginning with ABC that are of any object type for integrity violations. Objects with an incorrect domain, program and module objects that have been tampered with, and objects with not valid or missing digital signatures will cause integrity violation records to be logged to database file SECCHECK2. Database file SECCHECK2 will first be cleared of any existing records.

Example 4: Check Object in a Directory

```
CHKOBJITG OBJ('/PartOrder/Forms.jar')
  OUTFILE(SECCHECK3) OUTMBR(*FIRST *REPLACE)
  CHKDMN(*NO) CHKPGMMOD(*NO) CHKSIG(*ALL) CHKLIB(*NO)
```

This command checks file Forms.jar in directory PartOrder for integrity violations. If the file has a digital signature that is not valid or is capable of being signed and has no signature, an integrity violation record will be logged to database file SECCHECK3. Database file SECCHECK3 will first be cleared of any existing records.

Note: Any Java programs associated with this stream file will be checked for valid signatures as well.

Error messages for CHKOBJITG

*ESCAPE Messages

>> CPFA0AA

Error occurred while attempting to obtain space.

CPFA0A9

Object not found.

CPFA093

Name natching pattern not found.

CPF2122

Storage limit exceeded for user profile &1.

CPF22F0

Unexpected errors occurred during processing. <<

CPF22D9

No user profiles of specified name exist.

CPF2204

User profile &1 not found.

CPF2213

Not able to allocate user profile &1.

CPF222E

&1 special authority is required.

CPF222F

Command not run.

>> CPF4AAC

User profile &2. not processed.

CPF980B

Object &1 in library &2 not available. <<

CPF9860

Error occurred during output file processing.

>> CPF9873

ASP status is preventing access to object. <<

CHKOUT (Check Out) Command Description

CHKOUT Command syntax diagram

Purpose

The Check Out (CHKOUT) command checks out an object. The object is named on the OBJ parameter. A user profile is used to determine who is checking out the object.

When an object is checked out, other users can read and copy the object. Only the user who has the object checked out can change the object until it is checked in (see the Check In (CHKIN) command).

For more information about integrated file system commands, see the Integrated file system topic in the File systems and management category of the Information Center.

Restrictions:

1. Only documents within QDLS, and byte stream files can be checked out.
2. The user who submits this command must have *CHANGE authority to the object and at least execute (search) authority to the directory prefixes in the path.
3. Not all file systems will support the CHKOUT command.

Required Parameters

OBJ Specifies the name of the object to check out or a pattern for multiple objects.

The object path name can be either a simple name or a name that is qualified with the name of the directory in which the object is located. A pattern can be specified in the last part of the path name. An asterisk (*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes.

See path names for more information on specifying path names.

Example for CHKOUT

```
CHKOUT OBJ('MYDIR/FILE1')
```

This command checks out FILE1 in the directory, MYDIR, to the job's user profile owner.

Error messages for CHKOUT

*ESCAPE Messages

CPFA09C

Not authorized to object.

CPFA09D

Error occurred in program &1.

» CPFA09E

Object in use. «

CPFA0A1

An input or output error occurred.

CPFA0A3

Path name resolution causes looping.

CPFA0A7

Path name too long.

CPFA0A9

Object not found.

CPFA0AB

Object name not a directory.

CPFA0AD

Function not supported by file system.

CPFA0B2

No objects satisfy request.

CPFA0BE

&1 objects checked in. &2 objects failed.

CPFA0BF

&1 objects checked out. &2 objects failed.

CPFA0DA

Object name is a directory.

» CPFA1C5

Object is a read only object. «

CHKPWD (Check Password) Command Description

CHKPWD Command syntax diagram

Purpose

The Check Password (CHKPWD) command checks a password for the user running the command and determines its validity. The password is the security key that allows the user to sign on the system.

Required Parameter

PASSWORD

Specifies a password value that is checked for validity. If the password is correct, no message is shown. If the password not correct, one of the following messages is shown after each attempt:

CPF2362

Password not correct.

CPF2363

One attempt left to check your password.

CPF2364

The number of allowable attempts to check your password has been exceeded.

Example for CHKPWD

```
CHKPWD JOHNJONES
```

This command checks whether the current password is JOHNJONES.

Error messages for CHKPWD

*ESCAPE Messages

CPF2362

Password not correct.

CPF2363

Only 1 attempt left to check password.

CPF2364

Maximum number of attempts to check password reached.

CHKPRDOPT (Check Product Option) Command Description

CHKPRDOPT Command syntax diagram

Purpose

The Check Product Option (CHKPRDOPT) command reports differences between the correct structure and the actual structure of a software product. (For example, if an object is deleted from an installed product, running this command will report the error.) Use the informational and diagnostic messages to determine the condition of the product.

Note:

Running this command does not necessarily issue an escape message if the product has been deleted or is being created.

Required Parameter

PRDID

Specifies the identifier of the software product being checked.

***OPSYS:** The OS/400 licensed program is checked.

product-identifier: Specify a product identifier. The identifier must be seven characters in length.

Optional Parameters

RLS Specifies the release level of the product to be checked.

***ALL:** All releases of the product are checked.

***OPSYS:** The release level of the product being checked is the same as the release level of the operating system currently installed.

release-level: Specify the release level in the format VxRxMx, where Vx is the version number, Rx is the release number, and Mx is the modification level.

OPTION

Specifies the product option being checked.

***ALL:** All options of the product are checked.

***BASE:** The base option of the product is checked.

product-option: Specify an option number ranging from 1 through 99.

LODID

Specifies the product load being checked.

***ALL:** All product loads for a given option are checked.

***CODEDFT:** The code load is checked.

***PRIMARY:** The code load and the primary language load are checked.

product-load-identifier: Specify the product load identifier. The load identifier must be four characters in length.

» CHKSIG

Specifies if the digital signatures of objects are to be checked.

***SIGNED:** Objects with digital signatures are checked. Any object that has been signed will have the signature verified. Objects with signatures that are found to be not valid will be identified in messages sent to the job log and the product will be set to be in an erroneous state.

***ALL:** All objects that can be digitally signed will have the signature verified. Any object that can be signed but has no signature will be identified in a message sent to the job log but the product will not be set to be in error. Any signed object with a signature that is not valid will be identified in a message sent to the job log. If a signature is found to be not valid, the product will be set to be in an erroneous state.

***NONE:** Digital signatures of objects will not be checked.◀◀

DETAIL

Specifies which set of messages is sent for each product.

***BASIC:** Only the messages for the loads that actually exist are given. No messages are given for a load that is defined.

***FULL:** All messages are given for the loads requested.

Example for CHKPRDOPT

```
CHKPRDOPT PRDID(5716WP1)
```

This command checks all releases of the OfficeVision product with identifier 5716-WP1.

Error messages for CHKPRDOPT

*ESCAPE Messages

CPF0C20

Errors found by CHKPRDOPT.

» CPF0C2C

Errors found during digital signature verification.«

CPF0C4A

Product record not found.

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.

CPF0C54

Data in product record not correct.

CPF358A

Release not valid.

CPF8A06

Document &2 or folder &3 partially created in folder &1.

CPF8A78

Folder &1 in use.

CPF9012

Start of document interchange session not successful for &1.

CPF9032

Document interchange session not started.

CPF9830

Cannot assign library &1.

CPF9838

User profile storage limit exceeded.

CHKRCDLCK (Check Record Locks) Command Description

CHKRCDLCK Command syntax diagram

Purpose

The Check Record Locks (CHKRCDLCK) command detects whether a job has any record locks. This command is used to detect whether the job has any record locks before transferring to another group job while the user is in the middle of changing the database. This command can be used exclusively to check record locks on a job, and then the user can exit the command.

The CHKRCDLK command sends an escape message if there are any record locks currently held by the job that issued the command.

There are no parameters for this command.

Example for CHKRCDLCK

CHKRCDLCK

This command sends an escape message if there are any record locks held by the job.

Error messages for CHKRCDLCK

***ESCAPE Messages**

CPF321F

Job holds &1 record locks.

CHKTAP (Check Tape) Command Description

CHKTAP Command syntax diagram

Purpose

The Check Tape (CHKTAP) command searches a volume on the specified device for a unique volume identifier and/or file label. If the correct tape is on the device, the user may process this file on the next tape operation by specifying the same sequence number that was specified in the CHKTAP command. If the correct tape is not found, an escape message is sent, and the tape is rewound regardless of the end option specified. A check for this message in a CL program can direct the logic flow if the correct tape is on the device.

If a volume identifier is specified in the command, the volume identifier of the tape is compared with the volume identifier of the command. If a match is not found on the tape, an escape message is sent. This message contains the volume identifier found on the tape. If the correct volume identifier is found or if no volume identifier is specified in the command, and a sequence number is specified, that sequence number is located on the tape. For a standard labeled tape, the sequence number is determined from the header label of the file. For a tape that is not labeled, the sequence number is determined from the number of tape markers from the beginning-of-tape marker. If the sequence number is not found, an escape message is sent.

If the sequence number specified on the command is found and a label was specified, the file identifier in the header label is compared with the value in the LABEL parameter. The LABEL parameter is valid only for a standard-label tape. If the tape is a standard-label tape and the file label at that sequence number does not match the value in the LABEL parameter, an escape message is sent. This message contains the date the file was created and the file label for the file at that sequence number. If a match is found and a date is not specified in the command, the file is found. If the correct file identifier is found and a date is specified in the command, the date in the header label is compared with that of the command. If a match exists, the correct file is found. If there is not a match, an escape message is sent. This message contains the file label and the date that the file at that sequence number was created.

If SEQNBR (*SEARCH) is specified, a value for the LABEL parameter must be specified. The file label for each file on tape is checked for the LABEL parameter until a match is found. If the file is not found, an escape message is sent.

If the sequence number specified on the command is found and a LABEL parameter was not specified but the CRTDATE parameter was, the date in the header label for the file at the sequence number is compared with the date value in the CRTDATE parameter. If the dates do not match, an escape message is sent. The CRTDATE is valid only for standard label tape.

Note:

The values in the command parameters are compared with the values on the tape in the following order:

1. Volume identifier
2. Sequence number
3. File identifier in the header label at the sequence number specified
4. File creation date in the header label at the sequence number specified

The tape is checked for each parameter only if the parameters before it in the list are found on the tape or were not specified in the command.

Required Parameter

DEV Specifies the name of the device where the volume is being checked. Specify the name of the tape or media library device.

Optional Parameters

VOL Specifies one or more volume identifiers used by the file. More information on this parameter is in Commonly used parameters.

Note:

If the device specified is a media library device, then the volume specified should be the cartridge identifier to be mounted and used.

***MOUNTED:** The volume currently placed in the device is used. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

volume-identifier: If the device specified in the DEV parameter is a stand-alone tape device, specify the volume identifier of the labeled volume. The volume identifier read from the tape is compared to this value. If the volume identifier specified is not found on the tape, an escape message is sent. If the device specified in the DEV parameter is a library device description, specify the cartridge identifier of the volume to be used.

SEQNBR

Specifies whether a check is made for a specific sequence number of a data file on the tape. For standard-labeled tapes, this is the sequence number in the file header label. For tapes that are not labeled, this is the sequence number determined from the number of tape markers from the beginning of tape.

***NONE:** No check is made for a file on this volume.

***FIRST:** A check is made for the first file on this volume.

***NEXT:** A check is made for the next file on this volume. If the current sequence number is at the beginning of the volume, this value checks the first file on that volume.

***SEARCH:** A data file is searched for that has an identifier that matches the LABEL parameter value. If the last tape operation on the device specified ENDOPT(*LEAVE), the file search begins with the first data file beyond the current tape position. If ENDOPT(*LEAVE) was not used for the last tape operation, or if the tape was manually rewound since an ENDOPT(*LEAVE) operation, the search begins with the first data file on the volume. SEQNBR(*SEARCH) cannot be specified

when LABEL(*NONE) is also specified, and cannot be used to check a volume that is not labeled. An escape message is sent if the file is not found. If the sequence number is not found on the tape, an escape message is sent.

file-sequence-number: Specify the sequence number of the file that is used. Valid values range from 1 through 9999.

LABEL

Specifies whether a label identifier is checked. If a label is specified, a sequence number must be specified for the SEQNBR parameter.

***NONE**: No check is made for a label identifier on the tape. LABEL(*NONE) and CRTDATE(*NONE) must be specified for a volume that is not labeled.

file-label: Specify the identifier (17 characters maximum) of the data or save/restore file to check. If a label is specified, a sequence number must be specified for the SEQNBR parameter. The file identifier of the file at that sequence number is compared with the label identifier specified by this parameter. If the label does not match, a message is sent.

CRTDATE

Specifies whether the date on which the file was created is checked. If SEQNBR(*NONE) is specified, CRTDATE(*NONE) must also be specified.

***NONE**: The date on which the file was created is not checked. CRTDATE(*NONE) and LABEL(*NONE) must be specified for a volume that is not labeled.

file-creation-date: Specify the date that must match the date of the file being checked. The date must be entered in the format specified for the system values QDATFMT and, if separators are used, QDATSEP. If the date of the file being checked does not match the date specified by this parameter, an escape message is sent.

ENDOPT

Specifies the operation that is automatically performed on the tape volume after the operation ends. If more than one volume is included, this parameter applies only to the last tape volume used; all other tape volumes are rewound and unloaded when the end of the tape is reached.

***LEAVE**: The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

***REWIND**: The tape is rewound, but not unloaded.

***UNLOAD**: The tape is rewound and unloaded.

Examples for CHKTAP

Example 1: Checking the Volume Identifier

```
CHKTAP DEV(TAPE1) VOL(TAPEVOL)
```

This command checks the volume identifier of the volume on the tape device TAPE1. If the volume identifier on the tape is TAPEVOL, the command completes normally and no message is sent. If the volume identifier on the tape is not TAPEVOL, an escape message is sent.

Example 2: Checking for a Specific Sequence Number

```
CHKTAP DEV(TAPE2) VOL(VOLID) SEQNBR(5)  
LABEL(FILE5) CRTDATE('1/9/84')
```

This command checks the volume on the tape device TAPE2 for a volume identifier of VOLID. If that volume is found, sequence number 5 is located on the tape (it must be a standard-labeled tape). The sequence number in the file label is used to position to sequence number 5. If the sequence number is found and the header label contains both the file identifier FILE5 and the date of 1/9/84, the correct tape and file has been found, and a completion message is sent. The next tape operation can specify sequence

number 5 to access this file without positioning the tape. If the specified volume is not found or the tape is not a standard labeled volume, an escape message is sent. If the volume is found and the sequence number is not found, an escape message is sent. If the file label at that sequence number is not FILE5, an escape message is sent. If the date at that sequence number is not 1/9/84, an escape message is sent.

Error messages for CHKTAP

***ESCAPE Messages**

CPF6708

Command ended due to error.

CPF6718

Cannot allocate device &1.

CPF6720

Incorrect volume &2 found on device &1.

CPF6721

Device &1 not a tape device.

CPF6728

LABEL(*NONE) CRTDATE(*NONE) required for nonlabeled volume.

CPF6734

File sequence number &3 not found on volume &2.

CPF6735

Label ID &6 not found at &3.

CPF6736

Creation date &6 not found at &3.

CPF6737

Label &4 not found on volume &2.

CPF6745

Device &1 not a media library device.

CPF6751

Load failure occurred on device &4.

CPF6752

SEQNBR(*FIRST) or SEQNBR(*NEXT) is not valid.

CPF6760

Device &1 not ready.

CPF6772

Volume on device &1 cannot be processed.

CPF67E6

Volume &2 is not correct

CPF9814

Device &1 not found.

CPF9825

Not authorized to device &1.

CLRDKT (Clear Diskette) Command Description

CLRDKT Command syntax diagram

Purpose

The Clear Diskette (CLRDKT) command deletes all active and inactive files from a diskette by deleting the data file identifiers from the diskette label area. A single (expired) file is defined, covering the entire diskette, and is identified as DATA. The data contained in the files is not deleted. Refer to the DLTDKTLBL (Delete Diskette Label) command and the INZDKT (Initialize Diskette) command to delete the data in the files.

The CLRDKT command does not test the diskette for defects nor does it change the volume identifier and owner identifier fields. The error map also does not change.

The diskette in the specified device (DEV parameter) is cleared by the CLRDKT command. If no volume identifier is specified, the command clears the diskette in the specified device. If an identifier is specified and it is the same as the identifier on the diskette in the specified device, the diskette is cleared.

Note: When processing diskettes with labels that are not IBM standard labels, you may get unpredictable results. To initialize the diskette, enter the Initialize Diskette (INZDKT) command, with CHECK(*NO) specified.

Restriction: A diskette that has an extended label area cannot be cleared; it must be initialized by the Initialize Diskette (INZDKT) command.

Required Parameter

DEV Specifies the name of the device in which the diskette being cleared is located.

Optional Parameters

VOL Specifies one or more volume identifiers used by the file. More information on this parameter is in Commonly used parameters.

***MOUNTED:** The volume currently placed in the device is used.

volume-identifier: Specify a volume identifier to be compared with the diskette label volume identifier field of the diskette being cleared. The identifier can have up to 6 characters. Any combination of letters and numbers can be used.

CHECK

Specifies whether to check the diskette in the specified device for active files before it is cleared. Active files are files having an expiration date later than the system date.

***YES:** A check is performed on files whose labels are in cylinder 0 only. File labels in an extended file label area (not supported by the iSeries 400) are not checked. If any active files are found on a diskette, a message is sent to the system operator. The operator can continue the clear function, and erase the active files, or he can end the operation.

***NO:** The diskette is cleared without being checked for active files.

Examples for CLRDKT

Example 1: Clearing Diskette with Volume Identifier of MASTER

```
CLRDKT DEV(QDKT) VOL(MASTER) CHECK(*NO)
```

This command clears the diskette in device QDKT if its volume identifier is MASTER.

Example 2: Checking for Active Files Before Diskette is Cleared

CLRDKT DEV(DKT1)

This command clears the diskette in device DKT1. Because VOL(*MOUNTED) is assumed, the diskette could be in either the basic exchange or save/restore formats, and a volume identification check is not made. However, because CHECK(*YES) is also assumed, the diskette is checked for active files before it is cleared.

Error messages for CLRDKT

*ESCAPE Messages

CPF6156

Cancel reply received for message &6.

CPF6159

Clear diskette ended; previous errors occurred.

CPF9814

Device &1 not found.

CPF9825

Not authorized to device &1.

CLRJOBQ (Clear Job Queue) Command Description

CLRJOBQ Command syntax diagram

Purpose

The Clear Job Queue (CLRJOBQ) command removes all the batch jobs (including jobs that have been held) from the specified job queue. Any jobs that are currently being read in and interactive jobs that have been rerouted to the job queue remain on the queue. The running of jobs that were started from the job queue is not affected.

Required Parameter

JOBQ Specifies the qualified name of the job queue that is cleared of all waiting or held jobs.

The name of the job 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, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-queue-name: Specify the name of the job queue.

» LOG

Specifies whether to use the message logging values associated with a job for jobs removed from the job queue.

***JOB:** Use the message logging values specified for each job when the job is removed from the job queue.

***NONE:** No job log spooled files will be generated for the removed jobs. <<

Example for CLRJOBQ

```
CLRJOBQ JOBQ(QBATCH)
```

This command removes all jobs currently in the IBM-supplied job queue, QBATCH. Any job currently being read in is not affected.

Error messages for CLRJOBQ

*ESCAPE Messages

CPF2207

Not authorized to use object &1 in library &3 type *&2.

CPF3307

Job queue &1 in &2 not found.

CPF3330

Necessary resource not available.

CPF3416

&1 entries deleted. &2 entries not deleted from job queue &3 in library &4.

CPF9843

Object &1 in library &3 type &2 cannot be accessed.

CLRLIB (Clear Library) Command Description

CLRLIB Command syntax diagram

Purpose

The Clear Library (CLRLIB) command deletes all of the objects from the specified library that a user has the authority to delete. The CLRLIB command does not delete the specified library, only the objects for which the user has object existence authority; the other objects remain in the library. Objects being used by any other job when this command is entered are not deleted.

Restrictions:

1. The user must have object existence authority for every object being deleted and *USE authority to the library.
2. >> The user must have *USE authority to the auxiliary storage pool (ASP) device if a specific ASP device name is specified on the ASPDEV parameter. <<
3. >> This command cannot be used to clear the QRECOVERY, QRCYxxxxx, QSPL, QSPLnnnn, QSYS, QSYSxxxxx, QSYS2, QSYS2xxxxx, QSYSCGI, SYSIBM, or SYSIBxxxxx libraries (where 'xxxxx' is the number of a primary auxiliary storage pool (ASP) and 'nnnn' is the number of a basic user ASP.)<<
4. This command is conditionally threadsafe. The following restrictions apply:
 - a. In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA.

Required Parameter

LIB Specifies the name of the library that is cleared of all objects for which the user has object existence authority. Objects for which the user does not have object existence authority remain in the library.

***CURLIB:** The current library for the thread is cleared. If no library is specified as the current library for the thread, the QGPL library is used.

library-name: Specify the name of the library to be cleared.

» Optional Parameter

ASPDEV

Specifies the auxiliary storage pool (ASP) device name where storage is allocated for the library being cleared (LIB parameter). If the library is in an ASP that is not part of the thread's library name space, this parameter must be specified to ensure the correct library is cleared. ASPDEV(*) is the only valid value if ASPDEV is used when *CURLIB is specified for the LIB parameter.

*****: The ASPs that are currently part of the thread's library name space will be searched to locate the library. This includes the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32), and, if the thread has an ASP group, the primary and secondary ASPs in the ASP group.

***CURASGRP:** If the thread has an ASP group, the primary and secondary ASPs in the ASP group will be searched to locate the library. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched.

***SYSBAS:** The system ASP (ASP 1) and all defined basic user ASPs (ASPs 2-32) will be searched to locate the library. No primary or secondary ASPs will be searched, even if the thread has an ASP group.

auxiliary-storage-pool-device-name: The device name of the primary or secondary ASP to be searched to locate the library. The primary or secondary ASP must have been activated (by varying on the ASP device) and have a status of 'Active' or 'Available'. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched. <<

Example for CLRLIB

```
CLRLIB LIB(A)
```

This command deletes all objects in library A that are not in use and for which the user has object existence authority.

Error messages for CLRLIB

*ESCAPE Messages

» CPF8ED

Device description &1 not correct for operation. <<

CPF210D

Library &1 in use.

CPF2110

Library &1 not found.

CPF2113

Cannot allocate library &1.

CPF2129

Clear or delete of system library &1 canceled.

» CPF216B

Library &1 cannot be cleared. <<

CPF2161

Cannot delete some objects in library &1.

» CPF2173

Value for ASPDEV not valid with special value for library. «

» CPF218C

&1 not a primary or secondary ASP. «

CPF2182

Not authorized to library &1.

CPF8122

&8 damage on library &4.

» CPF9814

Device &1 not found. «

» CPF9825

Not authorized to device &1. «

» CPF9833 E

*CURASGRP or *ASPGRPPRI specified and thread has no ASP group. «

CLRMSGQ (Clear Message Queue) Command Description

CLRMSGQ Command syntax diagram

Purpose

The Clear Message Queue (CLRMSGQ) command clears (removes) all messages from a specified message queue. Once cleared, the data cannot be shown or printed. If the specified message queue is not allocated to a job, it is implicitly allocated by this command for the duration of the command. If the specified message queue is *WRKSTN or a work station message queue, it is not allocated and the message queue is cleared even if the work station device description is allocated to another job.

Required Parameter

MSGQ

Specifies the qualified name of the message queue that is cleared. If a specific message queue name is specified (instead of a generic name), only the first message queue found with that name is cleared.

***WRKSTN:** The work station message queue is cleared. This is not allowed in batch mode.

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, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue that is cleared.

Optional Parameter

CLEAR

Specifies which messages are cleared from the message queue.

***ALL:** All messages are cleared from the message queue. If there are unanswered messages on the queue, the default reply is returned to the sender before the message is cleared.

***KEEPUNANS:** Unanswered inquiry messages remain on the specified message queue. Other messages are removed.

Examples for CLRMSGQ

Example 1: Clearing All Messages

```
CLRMSGQ  MSGQ(*CURLIB/MQFIN)
```

This command clears all messages from a message queue named MQFIN, which is located in the current library for the job.

Example 2: Keeping Unanswered Messages

```
CLRMSGQ  MSGQ(*CURLIB/MQFIN)  CLEAR(*KEEPUNANS)
```

This command clears all messages except unanswered inquiry messages from a message queue called MQFIN, which is located in the current library for the job.

Error messages for CLRMSGQ

*ESCAPE Messages

CPF2357

Message queue &1 in &2 not cleared.

CLROUTQ (Clear Output Queue) Command Description

CLROUTQ Command syntax diagram

Purpose

The Clear Output Queue (CLROUTQ) removes spooled files from the specified output queue. The CLROUTQ command removes all spooled files that are waiting to be written on an output device from the specified queue, including files that are in the hold state. Spooled files that are currently being produced by programs or that are being written to an output device are not removed from the queue.

Required Parameter

OUTQ Specifies the qualified name of the output queue.

The name of the output 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, the QGPL library is used.

library-name: Specify the name of the library to be searched.

output-queue-name: Specify the name of the output queue that is cleared.

Example for CLROUTQ

```
CLROUTQ  OUTQ(QPRINT)
```

This command removes the entries for all spooled files from the output queue, QPRINT, that are waiting to be printed or are being held. The entries for the file currently being printed and files still receiving data from programs that are currently running are not affected.

Error messages for CLROUTQ

*ESCAPE Messages

CPF2207

Not authorized to use object &1 in library &3 type *&2.

CPF3330

Necessary resource not available.

CPF3357

Output queue &1 in library &2 not found.

CPF3417

&1 entries deleted. &2 entries not deleted.

CPF9843

Object &1 in library &3 type &2 cannot be accessed.

CLRPFM (Clear Physical File Member) Command Description

CLRPFM Command syntax diagram

Purpose

The Clear Physical File Member (CLRPFM) command removes all the data (including deleted records) from the specified member of a physical file. If ALLOCATE(*NO) is specified when the file is created, the record count for the member is set to zero, and the member size is set to the minimum size possible. If ALLOCATE(*YES) is specified when the file is created, the CLRPFM command resets the member size to the value used when the file is initially created. For more information, refer to the ALLOCATE parameter for the Create Physical File (CRTPF) command. An attempt to get a record from the cleared member results in an error message being sent to the user or program that made the attempt.

Note:

The CLRPFM command ignores all file overrides that are currently in effect for the job.

Restrictions:

1. The user must have object operational, object management or alter, and delete authority for the physical file that contains the member and execute authority to the library.
2. If any of the access paths to the member are in use when this command is entered, or if the physical file member is in use, the command is not run.
3. An *EXCL lock is required on the member to clear it.
4. In multithreaded jobs, this command is not threadsafe for distributed files. This command is also not threadsafe and fails for Distributed Data Management (DDM) files of type *SNA.

Required Parameter

FILE Specifies the qualified name of the physical file that contains the member being cleared.
The name of the physical 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.

physical-file-name: Specify the name of the physical file.

Optional Parameter

MBR Specifies the name of the member, or the first member, that is cleared.

***FIRST:** The first member in the database file is used.

***LAST:** The last member of the specified physical file is cleared.

physical-file-member-name: Specify the name of the physical file member that is cleared.

Example for CLRPFM

```
CLRPFM FILE(*CURLIB/INV) MBR(FEB)
```

This command clears the member named FEB in the physical file INV, found in the current library for the job *CURLIB. It is not cleared until all jobs currently using the member and all jobs using the access paths over the member are done with it.

Error messages for CLRPFM

*ESCAPE Messages

CPF3130

Member &2 already in use.

CPF3133

File &1 in library &3 contains no members.

CPF3134

Referential constraint error processing member &2.

CPF3136

File &1 in &3 not allowed on command.

CPF3137

No authority to clear, initialize, or copy member &2.

CPF3141

Member &2 not found.

CPF3142

File &1 in library &3 not found.

CPF3144

Member &2 not cleared or initialized.

CPF3156

File &1 in library &3 in use.

CPF3157

Triggers prevent requested operation.

CPF3159

Member &2 saved with STG(*FREE).

CPF3160

Operation on member &2 ended. Entry cannot be journaled.

CPF3179

Cannot clear or initialize DDM file &1 in &3.

CPF32B8

Distributed file error, reason code &3.

CPF32CF

Distributed file error, reason code &3.

CPF32C3

Distributed file error, level ID mismatch

CPF320B

Operation was not valid for database file &1.

CPF3203

Cannot allocate object for file &1 in &2.

CLRPOOL (Clear Pool) Command Description

CLRPOOL Command syntax diagram

Purpose

The Clear Pool (CLRPOOL) command clears all objects from a main storage pool. This allows the Set Object Access (SETOBJACC) command to report on storage usage within a pool.

Required Parameter

POOL Specifies the pool to be cleared of all objects.

***JOB:** The pool associated with the job is cleared.

***SHRPOOLn:** A general-purpose shared pool is cleared. Valid values range from 1 through 10.

Element 1: Subsystem

subsystem: Specify a subsystem name.

Element 2: Pool Identifier

pool-identifier: Specify a subsystem pool identifier.

Example for CLRPOOL

```
CLRPOOL POOL(*JOB)
```

This command clears the pool associated with the job in which the command was processed.

Error messages for CLRPOOL

***ESCAPE Messages**

CPF1858

The specified pool does not exist.

CPF1859

Use of an access path was requested but none exists.

CLRSVAF (Clear Save File) Command Description

CLRSVAF Command syntax diagram

Purpose

The Clear Save File (CLRSVAF) command clears the contents of a save file. This command clears all existing records from the save file and reduces the amount of storage used by this file.

A save file must be cleared before it can be used again to receive data from a save command or to receive another save file. If the user attempts to write new save data into a save file that already contains records, an inquiry message is sent to the work station for an interactive job, or to the system operator for a batch job, unless a save command is used and CLEAR(*ALL) is specified.

Note: This command ignores all file overrides that are currently in effect for the job.

Restriction: The user of this command must have operational and object management authority for the save file and read authority for the specified library.

Required Parameter

FILE Specifies the qualified name of the save file to clear.

The name of the save 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.

save-file-name: Specify the name of the save file to clear.

Example for CLRSVAF

```
CLRSVAF FILE(ONLINE)
```

This command clears the contents of save file ONLINE. Any existing records in the file are removed, and the file size is reduced to the minimum size possible.

Error messages for CLRSVAF

***ESCAPE Messages**

CPF3782

File &1 in &2 not a save file.

CPF3812

Save file &1 in &2 in use.

CPF9807

One or more libraries in library list deleted.

CPF9808

Cannot allocate one or more libraries on library list.

CPF9810

Library &1 not found.

CPF9812

File &1 in library &2 not found.

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

CPF9830

Cannot assign library &1.



CLRSVRSEC (Clear Server Security Data) Command Description

CLRSVRSEC Command syntax diagram

Purpose

The Clear Server Security Data (CLRSVRSEC) command clears decryptable authentication information that is associated with user profiles and validation list (*VLDL) entries. This is the same information that was cleared in releases previous to V5R2 when the QRETSVRSEC system value was changed from '1' to '0'.

There are no parameters for this command.

Restrictions:

- You must have *ALLOBJ and *SECADM special authorities to use this command.
- QRETSVRSEC system value must be '0'.

Example for CLRSVRSEC

```
CLRSVRSEC
```

This command checks that the QRETSVRSEC system value is set to '0' and, if so, clears decryptable authentication information.

Error messages for CLRSVRSEC

*ESCAPE Messages

CPF222E

&1 special authority is required.

CPF4AB4

QRETSVRSEC system value must be '0'.



CLRTRCDTA (Clear Trace Data) Command Description

CLRTRCDTA Command syntax diagram

Purpose

The Clear Trace Data (CLRTRCDTA) command clears (removes) all of the data from any previous trace operations in this debugging session. Once cleared, the data can no longer be displayed or printed.

Restriction: This command is valid only in debug mode. To start debug mode, refer to the STRDBG (Start Debug) command.

There are no parameters for this command.

Example for CLRTRCDTA

```
CLRTRCDTA
```

This command clears all of the data recorded from any and all previous tracing operations in all of the programs currently being debugged.

Error messages for CLRTRCDTA

*ESCAPE Messages

CPF1999

Errors occurred on command.

CLOF (Close File) Command Description

CLOF Command syntax diagram

Purpose

The Close File (CLOF) command closes a database file opened with the Open Database File (OPNDBF) or Open Query File (OPNQRYF) commands.

Restrictions:

1. This command is only used to close a file that is opened by the OPNDBF or OPNQRYF command.

Required Parameter

OPNID

Specifies the name used on the Open Query File (OPNQRYF) or the Open Database File (OPNDBF) for identifying this open identifier (OPNID). This OPNID is specified when closing this file. It cannot be reused without first closing this file.

Example for CLOF

```
CLOF OPNID(APPL)
```

This command closes a database file that was opened with APPL as the OPNID. The file was previously opened using the OPNDBF or OPNQRYF command with APPL specified (or defaulted) as the OPNID.

Error messages for CLOF

*ESCAPE Messages

CPF4519

Member &3 file &1 not closed.

CPF4520

No file open with identifier &4.

COMMIT (Commit) Command Description

COMMIT Command syntax diagram

Purpose

The Commit (COMMIT) command is used to complete the current transaction and to establish a new commitment boundary for the commitment definition associated with the program issuing the command.

The Start Commitment Control (STRCMTCTL) command must be issued first to establish the commitment definition before the COMMIT command is issued; otherwise, a message is sent.

When the COMMIT command is issued, all pending changes made to resources under commitment control for the commitment definition since the last commitment boundary was established are made permanent. A commitment identifier can be specified that is associated with this set of changes. If any files or API commitment resources associated with a journal are under commitment control, the commitment identifier is placed in the changes committed (CM) journal entry of each journal.

The commitment identifier is also used by the system when updating the notify object if it needs updating during activation group end, job end, or IPL (initial program load) processing.

No error occurs if there are no resources under commitment control for the commitment definition at the time the commit is issued. All record locks held for files opened under commitment control for the commitment definition are released when the commit is issued. Locks on object level commitment control resources, acquired when the resources are created or changed during the transaction are released when the commit is issued.

More information on commitment control is in the Commitment control article in the Information Center.

Optional Parameter

CMTID

Specifies the text used to identify a group of changes committed with the commitment boundary. This text is placed in the object specified on the NFYOBJ parameter of the STRCMTCTL command during IPL processing if an abnormal system failure occurs, or if a job ends with uncommitted changes or with a nonzero completion code.

***NONE:** No text is used to identify the group of changes committed with this commitment boundary.

***LUWID:** The logical unit of work identifier and the default journal name for this logical unit of work are used to identify the group of changes being committed with this commitment boundary.

'description': Specify a maximum of 3000 characters, enclosed in apostrophes, to identify the group of changes being committed with this commitment boundary.

Example for COMMIT

```
COMMIT CMTID('Account #123456 changes end')
```

This command specifies that all changes made to this point for the commitment definition associated with the program issuing the command are committed. The commitment identifier is Account #123456 changes end and may be used by the system when updating the notify object if it needs updating during activation group end, job end, or IPL processing.

Error messages for COMMIT

***ESCAPE Messages**

CPF5030

Partial damage on member &4.

CPF509F

Job has successfully connected after I/O error.

CPF5104

Cancel reply received for message &7.

CPF511D

Parameter integrity error occurred with reason code &1.

CPF5134

Not authorized to process request on member &4.

CPF5149

Operation for program device or member &4, file &2 in library &3 is not valid.

CPF5168

Cannot open member &3 file &1 in &2.

CPF5169

Cannot complete input or output (I/O) to DDM file &2 in &3.

CPF5173

&6 records in buffer not valid.

CPF5235

Entry for member &4 not journaled.

CPF5257

Failure for device or member &4 file &2 in library &3.

CPF5272

Records not added to member &4.

CPF83DB

Commit operation resulted in rollback.

CPF83D0

Commitment operation not allowed.

CPF83E1

Commit operation failed due to constraint violation.

CPF83E2

Rollback operation required.

CPF835F

Commit or rollback operation failed.

CPF8350

Commitment definition not found.

CPF8363

Commit operation failed.

CPF8367

Cannot perform commitment control operation.

CPF9203

Reply &1 received from DDM target system not expected.

CPF9255

Commitment control operation failed.

CMPJRNIMG (Compare Journal Images) Command Description

CMPJRNIMG Command syntax diagram

Purpose

The Compare Journal Images (CMPJRNIMG) command allows the user to compare and note the differences between (1) the before and after images of record-level changes (updates, deletes, rollback-updates, and rollback-deletes) for a specific file and/or member (IMAGES(*BOTH) must be specified for the Start Journal Physical File (STRJRNPf) command), or (2) the after and *previous after* image of a particular relative record (IMAGES(*AFTER) is specified for the STRJRNPf command). The output of the command is directed to a printer.

If before and after images are compared, the journaled changes can be compared for only one or all of the records in the specific file or member. The comparison can also be limited by a specific journal receiver range, or by a range of journal entries in a specific journal receiver range.

The printed output shows the record image before the change was made, followed by the record image after the change, followed by a line that indicates (with asterisks) the specific change in the record on a character-by-character basis, instead of on a field-by-field basis. If the journaled file has null-capable fields, the null value indicators that correspond to the before-image of the record are compared with the null value indicators that correspond to the after-image of the record. This is done on a field-by-field basis.

If there is no journal entry satisfying the search value specified, the command ends.

Restrictions:

1. The result of the comparison is sent only to the system printer.
2. The file/member specified must currently exist on the system and must have been journaled.
3. Only one member can be processed per command.
4. The comparison of journal images ends if one of the following conditions occurs:
 - The member was saved with storage freed.
 - The member was restored.
 - The member was cleared.
 - The member was initialized.
 - The member was reorganized.
 - The member was deleted.
 - The member was in use when the system ended abnormally.
 - Journaling the member was stopped.
 - The member had the journaled changes applied or removed (by the Apply Journaled Changes (APYJRNCHG) command or the Remove Journaled Changes (RMVJRNCHG) command).
5. If the sequence number is reset in the range of receivers specified, the first occurrence of FROMENT or TOENT is used if the parameters are specified.
6. The FROMENT and FROMTIME parameters are mutually exclusive, as are the TOENT and TOTIME parameters.

7. The JOB, PGM, and USRPRF parameters cannot be used to specify selection criteria if one or more journal receivers in the specified receiver range was attached to a journal that had a RCVSIZOPT or FIXLENTDA option specified that omitted the collection of that data.
8. This command cannot be used on or with a remote journal.
9. If this command is used to compare journal images for a file that contains any fields of data type BLOB (binary large object), CLOB (character large object), or DBCLOB (double-byte character large object), these fields are not included in the comparison. All other fields in the file are compared.
10. This command cannot be used if one or more journal receivers in the specified range was attached to a journal that had MINENTDTA (minimize entry specific data) specified for *FILE objects.

Required Parameter

FILE Specifies the qualified name of the physical database file for which the journaled record-level changes are compared.

The name of the physical 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.

physical-file-name: Specify the name of the physical file.

Optional Parameters

MBR Specifies the name of the member in the file that has its journal entries compared.

***FIRST:** The first member in the database file is used.

member-name: Specify the name of the member for which record-level changes are compared.

RCVRNG

Specifies the starting and ending journal receivers used in the comparison. The system starts the comparison with the starting journal receiver (specified by the first value) and proceeds through receivers until the ending journal receiver (specified by the last value) is processed. If dual receivers are used at any time, the first of the dual receivers is always used when chaining through the receivers.

If any problem (such as damaged receivers or receiver not found) occurs in the receiver chain before the comparison starts, the system tries to use the second of the dual receivers. If the second of the receivers is damaged or not found, or if a problem occurs during the operation, the comparison ends.

Note:

If the maximum number of receivers in the range is exceeded (256), an exception is sent and no entries are compared.

Single Value

***CURRENT:** The journal receiver that is currently attached when starting to compare journal entries is used.

Element 1: Starting Journal Receiver

The name of the journal receiver 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.

starting-journal-receiver: Specify the name of the first journal receiver that contains the journal entries that are compared.

Element 2: Ending Journal Receiver

***CURRENT:** The journal receiver that is currently attached when starting to compare journal entries is used.

The name of the journal receiver 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.

ending-journal-receiver: Specify the name of the last journal receiver that contains the journal entries to be compared.

FROMENT

Specifies the first journal entry to be compared.

***FIRST:** The first journal entry in the journal receiver range specified is the first entry considered for the comparison.

starting-sequence-number: Specify the sequence number at which the comparison begins.

FROMTIME

Specifies the date and time of the first journal entry to be compared. The journal entry with the specified date and time or the next later journal entry is the starting point for the comparison.

Element 1: Date When Comparison Starts

starting-date: Specify the date on which comparison of the first entry starts.

Element 2: Time When Comparison Starts

starting-time: Specify the time at which comparison of the first entry starts. The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits, where the time separator for the job separates the hours, minutes, and seconds. If you issue this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

TOENT

Specifies the last entry considered for the comparison.

***LAST:** The last journal entry in the journal receivers specified is the final entry compared.

ending-sequence-number: Specify the sequence number of the last journal entry to be compared.

Note:

The values specified for the FROM and TO parameters can be the same (for example, FROMENT(234) and TOENT(234) can be specified).

TOTIME

Specifies the date and time of the last journal entry to be compared. The journal entry with the specified date and time or the latest earlier journal entry is the ending point for the comparison of journal entries.

Element 1: Date When Comparison Ends

ending-date: Specify the date on which the comparison ends.

Element 2: Time When Comparison Ends

ending-time: Specify the time at which the comparison ends. The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits, where the time separator for the job separates the hours, minutes, and seconds. If you issue this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

CMPOPT

Specifies the type of record images that are compared for record-level changes in the specified file.

***BOTH:** The before-images of the journal entries are compared with the after-images of the journal entries.

***AFTER:** The after-images of the record (specified in the RCDNBR parameter) are compared with previous after-images.

Note:

If this value is specified, (1) the default value *ALL must be specified on the JOB, PGM, USRPRF, and CMTCYCID parameters and (2) a relative record number must be specified on the RCDNBR parameter.

RCDNBR

Specifies the relative record number in the file for which the journal entry images are compared.

***ALL:** The journaled changes for all records in the physical member are compared.

relative-record-number: Specify the relative record number in the physical member for which images are compared. If a value is specified, only changes for the specified record are compared.

JOB Specifies that the comparison is of journal entries for a particular job.

A job identifier is either the special value *ALL or a qualified name with up to three elements. For example:

*ALL
job-name
user-name/job-name
job-number/user-name/job-name

More information on this parameter is in Commonly used parameters.

***ALL:** The comparison is not limited to entries for a particular job.

job-name: Specify the name of the job whose journaled changes are considered for comparison.

user-name: Specify the name of the user of the job whose journaled changes are considered for comparison.

job-number: Specify the number of the job whose journaled changes are considered for comparison.

PGM Specifies that the comparison is of journal entries for a particular program.

***ALL:** The comparison is not limited to entries for a particular program.

program-name: Specify the name of the program whose record-level changes are to be considered for comparison. Only changes journaled for this program are considered for comparison.

USRPRF

Specifies that the comparison is of journal entries for a particular user profile name. The user profile name is the user profile under which the job is run that causes the entries to be journaled.

***ALL:** The comparison is not limited to entries for a particular user profile.

user-name: Specify the name of the user profile that has journaled changes to be compared. Only journaled changes for this user profile are to be considered for comparison.

CMTCYCID

Specifies the commit cycle identifier of the specific journal that participated in a logical unit of work for which a comparison of journal entries is made.

***ALL:** The journal entries for all commit cycle identifiers are included in the comparison.

commit-cycle-identifier: Specify the identifier for the commit cycle whose journaled changes are to be considered for comparison. A journal entry's commit cycle identifier can be found by using the Display Journal (DSPJRN) command and selecting option five.

OUTFMT

Specifies the format in which the objects are shown.

***CHAR:** The record images are shown in character format.

***HEX:** The record images are shown in hexadecimal format.

Examples for CMPJRNIMG

Example 1: Comparing Before-Images with After-Images

CMPJRNIMG FILE(QGPL/PF)

This command compares the journaled record-level changes for the first member of file PF in the QGPL library. The entries compared are in the journal receiver that is currently attached when the comparison begins, starting with the first entry and ending with the last entry. All entries with both before-images and after-images that satisfy the selection values are eligible to be compared. The before-images of the entries are compared with the after-images of the entries.

Example 2: Comparing After-Images with Previous After-Images

```
CMPJRNIMG FILE(MYLIB/PAYROLL) MBR(APRIL)
RCVRNG((RCVLIB/RCV3) (*CURRENT)) FROMENT(200)
TOENT(500) CMPOPT(*AFTER) RCDNBR(999) OUTFMT(*HEX)
```

This command compares the journaled record-level changes for the member named APRIL in file PAYROLL in MYLIB, beginning with receiver RCV3 in RCVLIB and ending with the journal receiver that is currently attached at the start of the comparison. The range of entries compared starts with entry 200 and ends with entry 500. Only the after-images and previous after-images are compared. The comparison is limited to record number 999. The output is printed in hexadecimal format.

Example 3: Specifying Journal Entry Date and Time

```
CMPJRNIMG FILE(USERLIB/MYFILE) MBR(*FIRST)
RCVRNG((RCV2) (USERLIB/RCV5))
FROMTIME('7/04/87' 120000) TOENT(1000)
```

This command compares the journaled record-level changes for the first member of file MYFILE in USERLIB, beginning with receiver RCV2 in *LIBL and ending with receiver RCV5 in USERLIB. The date and time of the first journal entry to be compared is 7/4/87 12:00:00, and the ending record sequence number considered for the comparison is 1000.

Error messages for CMPJRNIMG

*ESCAPE Messages

CPF7002

File &1 in library &2 not a physical file.

CPF7006

Member &3 not found in file &1 in &2.

CPF701B

Journal recovery of an interrupted operation failed.

CPF7027

Operation cannot be performed beyond entry &1.

CPF7028

Member &3 file &1 in &2 never journaled.

CPF7029

Image comparison failed. Ending sequence number &1.

CPF7036

File &1 in &2 not journaled with before images.

CPF7038

No entries compared for member &3.

CPF705A

Operation failed due to remote journal.

CPF7053

Values for RCVRNG parameter not correct; reason code &1.

CPF7054

FROM and TO values not valid.

CPF709C

JOB, PGM, and USRPRF not valid for receiver range.

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9803

Cannot allocate object &2 in library &3.

CPF9810

Library &1 not found.

CPF9812

File &1 in library &2 not found.

CPF9815

Member &5 file &2 in library &3 not found.

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9850

Override of printer file &1 not allowed.

CPROBJ (Compress Object) Command Description

CPROBJ Command syntax diagram

Purpose

The Compress Object (CPROBJ) command compresses programs, panel groups, menus, display files, printer files, modules, and service programs.

- *Compressed Objects* are objects that consume less storage space than decompressed objects. When a compressed object is used or a compressed program is called, a decompressed version of the object automatically becomes available to the user.
- *Decompressed Objects* are objects that use the system storage space allocated to them and are in a final, ready-to-use state.
- *Temporarily Decompressed Objects* are temporarily decompressed copies of compressed objects. The system allocates storage space for the temporary copies until the system or the user determines that the temporary storage space needs to be reclaimed.

Temporary storage is automatically reclaimed when:

- The RCLTMPSTG command is run
- The next initial program load (IPL) is run
- The object is used often enough to cause the system to permanently decompress it

When an object is permanently decompressed, the compressed version of the object is destroyed as well as any temporary forms of the object; however, compressed versions remain intact as long as the objects are temporarily decompressed.

Restrictions:

1. The user must have *OBJMGT authority to the object specified and *EXECUTE authority to the library containing the object.
2. Objects that were saved with storage freed cannot be compressed or decompressed.
3. Programs without a valid validation value are not compressed.
4. A program, service program, or module that was created prior to Version 3, Release 6 must be retranslated before the object can be compressed. Retranslate the object using the CHGPGM, CHGSRVPGM, or CHGMOD commands.
5. To compress a system program, the user must end all active subsystems.
6. To prevent abnormal end of a program, the program must not be running in the system when it is compressed.

Required Parameters

OBJ Specifies the qualified name of the object to be compressed.

The name of the object 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.

***USRLIBL:** Only the libraries in the user portion of the job's library list are searched.

***ALL:** All libraries in the system, including QSYS, are searched.

» ***ALLUSR:** User libraries are all libraries with names that do not begin with the letter Q except for the following:«

#CGULIB	#DSULIB	#SEULIB
#COBLIB	#RPGLIB	
#DFULIB	#SDALIB	

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

QDSNX	» QSYS2xxxx«	QUSROND
QGPL	QS36F	QUSRPOSGS
QGPL38	QUSER38	QUSRPOSSA
QMPGDATA	QUSRADSM	QUSRPYMSVR
QMQMATA	QUSRBRM	QUSRDRARS
QMQMPROC	QUSRDIRCL	QUSRSYS
QPFRRDATA	QUSRDIRDB	QUSRVI

QRCL
» QRCLxxxx«
» QSYS2«

QUSRIJS
QUSRINFSKR
QUSRNOTES

QUSRVxRxMx

Notes:

1. » 'xxxx' is the number of a primary auxiliary storage pool.«
2. A different library name, of the form QUSRVxRxMx, can be created by the user for each release that IBM supports. VxRxMx is the version, release, and modification level of the library.

library-name: Specify the name of the library to be searched.

***ALL**: All objects in the specified library, that can be compressed, are compressed.

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 compressed only if *ALL or *ALLUSR library values can be specified for the name. See generic names for additional information.

object-name: Specify the name of the object that is compressed.

OBJTYPE

Specifies the type of the object to be compressed.

***ALL**: All menus, panel groups, display and printer device files, programs, modules, and service programs with the specified object name are compressed.

***FILE**: The file with the specified object name is compressed (display and printer files only).

***MENU**: The menu with the specified object name is compressed.

***MODULE**: The module with the specified object name is compressed.

***PGM**: The program with the specified object name is compressed.

***PNLGRP**: The panel group with the specified object name is compressed.

***SRVPGM**: The service program with the specified object name is compressed.

Optional Parameters

DAYS Specifies the number of days an object has not been used or changed. If the object has not been used or changed for more than the specified number of days, it is compressed. If it has been used or changed, it is left decompressed.

***NONE**: The object is compressed regardless of the number of days it has not been used or changed.

number-of-days: Specify the number of days an object must be unused for it to be compressed. Valid values range from 1 through 366 days.

PGMOPT

Specifies, for *PGM or *SRVPGM objects, the program option that indicates whether the entire program (instruction stream and observability tables) or only the observability tables are compressed.

***ALL:** The entire program or service program is compressed.

***OBS:** Only the observability tables of the program or service program are compressed.

Example for CPROBJ

```
CPROBJ OBJ(QGPL/*ALL)
      OBJTYPE(*FILE)
```

Error messages for CPROBJ

*ESCAPE Messages

CPF2110

Library &1 not found.

CPF2113

Cannot allocate library &1.

CPF2176

Library &1 damaged.

CPF3B01

Cannot compress or decompress object &1 in &2.

CPF3B02

Cannot compress or decompress file &1 in &2.

CPF3B03

No objects compressed.

CPF3B04

&1 objects compressed; &3 not compressed; &8 not included.

CPF3B08

Cannot allocate object &1 in &2.

CPF3B09

Not all subsystems ended.

CPF3B10

Cannot compress object &1 in &2 type *&3.

CPF3B11

Cannot compress object &1 in &2 type *&3.

CPF8108

Device file or save file &4 in &9 damaged.

CPF812E

Module &4 in &9 damaged.

CPF8129

Program &4 in &9 damaged.

CPF813D

Service program &4 in &9 damaged.

CPF815D

*M36 object &4 in &9 damaged.

CPF815E

*M36CFG object &4 in &9 damaged.

CPF8150

Panel group &4 in &9 damaged.

CPF8151

Menu &4 in &9 damaged.

CPF9570

Error occurred creating or accessing debug data.

CPF9802

Not authorized to object &2 in &3.

CPF9803

Cannot allocate object &2 in library &3.

CPF9804

Object &2 in library &3 damaged.

CPF9806

Cannot perform function for object &2 in library &3.

CPF9807

One or more libraries in library list deleted.

CPF9808

Cannot allocate one or more libraries on library list.

CPF9811

Program &1 in library &2 not found.

CPF9812

File &1 in library &2 not found.

CPF9821

Not authorized to program &1 in library &2.

CPF9822

Not authorized to file &1 in library &2.

CPF9838

User profile storage limit exceeded.

CFGDEVMLB (Configure Device Media Library) Command Description

CFGDEVMLB Command syntax diagram

Purpose

The Configure Device Media Library (CFGDEVMLB) command connects the media library device description with the communication interface for media library devices that require a communication interface. The CFGDEVMLB command will configure the necessary communication information based on the input to the command. It will update the necessary information in the device description specified, and will attempt to vary on the media library device description. Refer to the Use tape library article in the Information Center for more information about configuring media library devices.

For a LAN-attached media library device, the information on the Library Manager console must also be updated. To determine the necessary information for the Library Manager, use the Display LAN Media Library (DSPLANMLB) command.

This command must be issued once for each media library device description that uses a communication interface.

User *PUBLIC will be given *USE authority to any objects that this command creates: controller, device, etc. The objects that the CFGDEVMLB command creates will be named the same as the resource name

specified on the command for ADPTTYPE(*RS232) and same as the remote location name for ADPTTYPE(*LAN). The owner of the objects will be the user profile of the user executing the command.

Restrictions:

- *IOSYSCFG special authority is required to run this command.
- *USE authority is also required for the following CL commands: CRTDEVAPPC, CRTCTLAPPC, CRTDEVASC, CRTCTLASC, CRTLINASC, CHGDEVMLB, and VRYCFG.

Required Parameters

DEV Specifies the name of the media library device. The device description must exist on the system whether it was autoconfigured, or it was created with the Create Device Media Library (CRTDEVMLB) command.

ADPTTYPE

Adapter type

***RS232:** Indicates that the media library device is attached with a RS-232 port.

***LAN:** Indicates that the media library device is attached with a token-ring or ethernet local area network line.

Optional Parameters

RSRCNAME

Specifies the resource name of the RS-232 port. Use the Work with Hardware Resources (WRKHDWRSC) command with TYPE(*CMN) to determine what resources exist on the system.

This parameter is required when ADPTTYPE(*RS232) is specified.

» PROTOCOL

Specifies the communication protocol to use to communicate with the robot.

This parameter is required when ADPTTYPE(*LAN) is specified.

***APPC:** Indicates the APPC protocol will be used to communicate with the robot.

***TCP:** Indicates the TCP/IP protocol will be used to communicate with the robot. «

LIND Specifies the line description name to which the media library device is attached. The line description must already exist on the system. Use the Work with Configuration Status (WRKCFGSTS) command, with CFGTYPE(*LIN), to display a list of line descriptions that are configured on the system.

This parameter is required when ADPTTYPE(*LAN) » and PROTOCOL(*APPC) are « specified. A maximum of 2 line descriptions can be specified.

RMTLOCNAME

Specifies the name of the Library Manager to which the media library device will communicate using the format nnnnnnnn.cccccc, where nnnnnnnn is the remote network identifier (ID) and cccccc is the remote location name. If no network ID is specified, the network attributes are used to determine the default network ID.

This parameter information should be obtained from the Library Manager console. To determine the remote location name on the Library Manager, select COMMANDS from the action bar of the MAIN MENU. From the COMMANDS pull-down, select LM LAN Options, and then select LM LAN Information. The LM LAN Information panel will display the correct location name and network identifier for this media library device.

This parameter is required when ADPTTYPE(*LAN) » and PROTOCOL(*APPC) are « specified. A maximum of 2 line descriptions can be specified.

ADPTADR

Specifies the LAN adapter address of the remote controller from the Library Manager.

This parameter information should be obtained from the Library Manager console. To determine the adapter address on the Library Manager, select COMMANDS from the action bar of the MAIN MENU. From the COMMANDS pull-down, select LM LAN Options, and then select LM LAN Information. The LM LAN Information panel will display the correct adapter address for this media library device.

This parameter is required when ADPTTYPE(*LAN) >> and PROTOCOL(*APPC) are << specified. A maximum of 2 line descriptions can be specified.

>> ROBOTHOST

Specifies the TCP/IP host name or internet address of the robotic library manager.

robot-host-name: The specified name of the robotic library manager. The user may enter the host name by entering the robot host name or the domain qualified robot host name. The domain qualified robot host name allows input of 255 bytes.

robot-internet-address: The specified address of the TCP/IP interface.

The robot host internet address must be of the form ddd.ddd.ddd.ddd where ddd is a decimal number ranging from 0 to 255 and should not contain leading zeroes.

This parameter is required when ADPTTYPE(*LAN) and PROTOCOL(*TCP) are specified. A maximum of 2 robot host names or robot internet addresses can be specified.

LCLINTNETA

Specifies the local internet address of the interface that is connecting to the robot library manager. This is the interface the operating system will start when TCP/IP needs to be started to use the media tape library.

local-ip-address: Specify the local internet address to be started.

The internet address must be of the form ddd.ddd.ddd.ddd where ddd is a decimal number ranging from 0 to 255 and should not contain leading zeroes.

This parameter may only be specified when ADPTTYPE(*LAN) and PROTOCOL(*TCP) are specified. <<

Examples for CFGDEVMLB

Example 1: Configuring a RS-232 attached media library device

```
CFGDEVMLB MLB(TAPLIB01) ADPTTYPE(*RS232) RSRNAME(CMN01)
```

This command will create the necessary RS-232 communication line, controller, and device and change the necessary parameters in the media library device description. It will also attempt to vary on the media library device. The command does this in the following order:

- Create Line Description (Async) - CRTLINASC CMN01 with a resource name of CMN01.
- Create Controller Description (Async) - CRTCTLASC CMN01.
- Create Device Description (Async) - CRTDEVASC CMN01.
- Change Device Description - CHGDEVMLB TAPLIB01 to change the parameter robot device to ROBOTDEV(CMN01) and to change the parameter online at IPL to ONLINE(*YES).
- Vary Config Description -VRYCFG TAPLIB01 to vary on the media library device.

If any of these commands is not successful, the CFGDEVMLB command will not be successful. Note that multiple media library device descriptions could have the same communication line. In this case, the line description, controller, and device will not be recreated, but the CHGDEVMLB and VRYCFG commands will still be used.

Example 2: Configuring a LAN attached media library device

```
CFGDEVMLB DEV(TAPLIB02) ADPTTYPE(*LAN) LIND(TRNLIN)
RMTLOCNAME(APPN.MLD01) ADPTADR(0123456789AB)
```

This command will create the necessary LAN communication controller and device and change the necessary parameters in the media library device description. It will also attempt to vary on the media library device. Note that the line description must exist prior to using the CFGDEVMLB command. The command does this in the following order:

- Create Controller Description (APPC) - CRTCTLAPPC MLD01 with ONLINE(*YES).
- Create Device Description (APPC) - CRTDEVAPPC MLD01.
- Vary Config Description -VRYCFG MLD01 to vary on MLD01 APPC controller that was created.
- Change Device Description - CHGDEVMLB TAPLIB02 to change the parameter robot device to ROBOTDEV(MLD01) and to change the parameter online at IPL to ONLINE(*YES).
- Vary Config Description -VRYCFG TAPLIB02 to vary on the media library device.

If any of these commands is not successful, the CFGDEVMLB command will not be successful. Note that multiple media library device descriptions could have the same communication line. In this case, the controller description, and device will not be recreated, but the CHGDEVMLB and VRYCFG commands will still be used.

Example 3: Configuring a LAN attached media library device with two remote locations >>

```
CFGDEVMLB DEV(TAPLIB02) ADPTTYPE(*LAN) PROTOCOL(*APPC)
LIND(TRNLIN) RMTLOCNAME(APPN.MLD01A APPN.MLD01B)
ADPTADR(0123456789AB 0123456789CD)
```



This command will create the necessary LAN communication controllers and devices and change the necessary parameters in the media library device description. It will also attempt to vary on the media library device. Note that the line description must exist prior to using the CFGDEVMLB command. The command does this in the following order:

- Create Controller Description (APPC) - CRTCTLAPPC MLD01A with ONLINE(*YES).
- Create Device Description (APPC) - CRTDEVAPPC MLD01A.
- Vary Config Description -VRYCFG MLD01A to vary on MLD01A APPC controller that was created.
- Create Controller Description (APPC) - CRTCTLAPPC MLD01B with ONLINE(*YES).
- Create Device Description (APPC) - CRTDEVAPPC MLD01B.
- Vary Config Description -VRYCFG MLD01B to vary on MLD01B APPC controller that was created.
- Change Device Description - CHGDEVMLB TAPLIB02 to change the parameter robot device to ROBOTDEV(MLD01A MLD01B) and to change the parameter online at IPL to ONLINE(*YES).
- Vary Config Description -VRYCFG TAPLIB02 to vary on the media library device.

If any of these commands is not successful, the CFGDEVMLB command will not be successful. Note that multiple media library device descriptions could have the same communication line. In this case, the controller description, and device will not be recreated, but the CHGDEVMLB and VRYCFG commands will still be used.

>> Example 4: Configuring a LAN attached media library device to communicate using TCP/IP

```
CFGDEVMLB DEV(TAPLIB02) ADPTTYPE(*LAN) PROTOCOL(*TCP)
ROBOTHOST(MLD01A) LCLINTNETA(10.1.2.3)
```

This command will change the robot information in the device description to the TCP/IP information provided. It will also attempt to vary on the media library device. Note that TCP/IP should be configured prior to using the CFGDEVMLB command. The command does this in the following order:

- Change Device Description - CHGDEVMLB TAPLIB02 to change the parameter robot host to ROBOTHOST(MLD01A) and the internet address to LCLINTNETA(10.1.2.3).
- Vary Config Description -VRYCFG TAPLIB02 to vary on the media library device.

If any of these commands is not successful, the CFGDEVMLB command will not be successful. <<

Error messages for CFGDEVMLB

*ESCAPE Messages

CPF222E

&1 special authority is required.

CPF6708

Command ended due to error.

CPF672B

Resource &1 not valid.

CPF672C

Device &1 not allowed.

CPF672D

Network ID &1 not in correct format.

CPF672E

Line description &2 wrong type.

CPF672F

Resource &1 not found.

CPF6745

Device &1 not a media library device.

CPF67E5

Local area network information not valid.

CPF9814

Device &1 not found.

CPF9825

Not authorized to device &1.

CFGDSTSRV (Configure Distribution Services) Command Description

CFGDSTSRV Command syntax diagram

Purpose

The Configure Distribution Services (CFGDSTSRV) command changes the configuration of the distribution network. The user can add, change, remove, and display entries from the distribution queues table, the routing table, and the secondary system name table. A detailed description of configuring a distribution

network is in the SNA Distribution Services  book.

Restrictions:

1. This command is shipped with public *EXCLUDE authority and the QPGMR and QSYSOPR user profiles have private authorities to use the command.

2. Before this command is run for the first time, the QSNADS subsystem must be started to create the internal systems network architecture distribution services (SNADS) objects that this command uses.
3. Messages that report errors about system names or distribution queues may show or print different characters than the user entered because of internal system transformations. The internal value for a system name or distribution queue may differ from the characters shown by the CFGDSTSRV command depending on the language being used for the work station.


Optional Parameter

OPTION

Specifies an option from the Distribution Services Menu display that bypasses the initial menu and goes directly to the indicated table. The user can specify the distribution queues table, routing table, or secondary system name table without showing the Distribution Services Menu display.


***SELECT:** The option is selected from the Distribution Services Menu display.

1: The distribution queues table function, which identifies all the distribution queues for systems adjacent to the local system, is used.

An example of the distribution queues function is in the SNA Distribution Services  book.

2: The routing table function, which describes explicit or default entries for the destination systems in the SNADS network to which distribution queue entries can be routed, is used. An example of

the routing table function is in the SNA Distribution Services  book.

3: The secondary system name function, which lists all names by which the system is known, is used. An example of the secondary system name function is in the SNA Distribution Services  book.

Example for CFGDSTSRV

```
CFGDSTSRV OPTION(1)
```

This command shows the distribution queues' table entries.

Configuration changes may be made to existing distribution queues, or additional distribution queues may be configured.

Error messages for CFGDSTSRV

*ESCAPE Messages

CPF8802

Distribution queue &1 was not found.

CPF8805

Special value for System name/Group not permitted or not used correctly.

CPF8806

Value &1 not valid for system name or system group.

CPF8807

Error occurred while using QSNADS journal.

CPF8809

Errors detected on SNADS internal queues.

CPF8814

Queue &1 not found.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

Error occurred while closing file &1 in library &2.

CPF9850

Override of printer file &1 not allowed.

CPI8854

DSNX error while journaling.

CFGHTTPSCH (Configure HTTP Search) Command Description

Note: To use this command, you must have the 5722-DG1 (HTTP Server) product installed.

Purpose

You can use the configure HTTP search (CFGHTTPSCH) command to perform various search administration tasks.

You can create an index, add documents to an index, remove documents from an index, delete an index, create, update, or delete a document list, and create or update a mapping rules file.

Restriction: You must have *IOSYSCFG special authority to use this command.

Required Parameter

Option

Specifies the administrative task to perform.

The possible values are:

***CRTIDX**

Create an index.

***MRGIDX**

Merge an index after documents have been added.

***DLTIDX**

Delete an index.

***ADDDOC**

Add documents to an index.

***RMVDOC**

Remove documents from an index.

***CRTDOCL**

Create a document list. If the file already exists, it will be replaced.

***UPDDOCL**

Append additional document paths to a document list.

***DLTDOCL**

Delete a document list.

***CRTMAPF**

Create a mapping rules file. If the file already exists, it will be replaced.

***UPDMAPF**

Append additional configuration directives to a mapping rules file.

Optional Parameters

IDX Specifies the name of the index.

IDXDIR

Specifies the index directory that is used for several files created during index administration.

The possible values are:

’/QIBM/USERDATA/HTTPSVR/INDEX’

This directory is used for the index directory.

’index-directory-name’

The index directory path name.

TEXT Specifies the text that describes the index.

The possible values are:

***BLANK**

No text is specified.

’description-of-index’

Specify the text enclosed in apostrophes.

DOCLIST

Specifies the name of the document list file that contains a list of the documents to be indexed.

The possible values are:

’document-list-file-path-name’

The document list file path name.

CONTENT

Specifies the contents of the documents to be indexed.

The possible values are:

***HTML**

Documents contain HTML tags. All HTML tags are removed during indexing. See also the IDXHTML parameter.

***TEXT** Documents contain text only.

ENBCASE

Specifies whether a case-sensitive search is allowed for this index.

The possible values are:

***YES** A case-sensitive search is allowed.

***NO** Only case-insensitive searches are allowed.

ALWCHAR

Specifies the characters that are valid for a search on this index.

The possible values are:

***ALPHANUM**

Alphanumeric characters are valid.

***ALPHA**

Only alphabetic characters are valid.

IDXHTML

Specifies the HTML tags that are used to find additional character strings to index. If *NONE is specified, all HTML tags are removed from the document before indexing. All searches will be done on the entire document.

Any tag field that is selected will be indexed separately and will also be included in the indexing of the entire document. Tagged fields or the entire document can be selected for a search.

This parameter is ignored unless CONTENT(*HTML) is also specified.

The possible values are:

***NONE**

All of the document is indexed except for HTML tags.

***TITLE**

Index the title field.

***ABSTRACT**

Index the META tag NAME="Abstract".

***AUTHOR**

Index the META tag NAME="Author".

***DESCRIPTION**

Index the META tag NAME="Description".

***KEYWORDS**

Index the META tag NAME="Keywords".

***ALLMETA**

Index the META tag NAME="xxxxx".

CONTENT(HTML)

Specifies whether to index tagged fields; otherwise the fields are ignored. The selected fields are indexed when the index is updated with HTML files.

ALWERR

Specifies whether to skip document file errors and continue processing the request or to stop processing on a document file error.

The possible values are:

***YES** Allow file errors and continue processing.

***NO** Do not allow file errors. Stop indexing the documents.

STRDIR

Specifies the starting directory to use to find documents to add to the document list.

The possible values are:

'starting-directory-name'

The directory to use to find documents to index.

SUBTREE

Specifies whether to traverse subdirectories of the starting directory when building the document list file.

The possible values are:

***ALL** Search the subdirectories.

***NONE**

Do not search the subdirectories.

PATTERN

Specifies the pattern or filter to use when building the document list. For example, use the filter *.htm* to find HTML files.

The possible values are:

.HTM

This filter will find files with the extension .HTM or .HTML.

'filter-pattern'

The pattern or filter to use for selecting files to add to the document list.

DLTTYPE

Specifies whether to delete all of the index or only the supplemental index. The supplemental index is temporarily created when new or modified documents are added to the index.

The possible values are:

***ALL** Delete the main and supplemental index.

>

***SUPP**

Delete only the supplemental index.

MAPFILE

Specifies the name of the mapping rules file that contains configuration information to use for creating URLs for documents found on a search.

The possible values are:

'mapping-rules-file-path-name'

Specifies the name of the mapping rules file.

CFG Specifies the configuration file that contains configuration directives.

The possible values are:

'configuration-file-name'

Specifies the name of the configuration file to use.

URLPFX

Specifies the prefix to use, for the URL, for documents found on a search.

The possible values are:

***NONE**

No URL prefix is used.

'URL-prefix'

Specifies the prefix to use for the URL such as 'http://www.mysys.com'.

Examples for CFGHTTPSCH

Example 1: Create a document list

```
CFGHTTPSCH OPTION(*CRTDOCL)
DOCLIST('/QIBM/USERDATA/HTTPSVR/+
INDEX/myindex.DOCUMENT.LIST')
STRDIR('/QIBM/ProdData/HTTP/Public/HTTPSVR/HTML')
```

This example will create a document list called /QIBM/USERDATA/HTTPSVR/INDEX/myindex.DOCUMENT.LIST from the directory /QIBM/ProdData/HTTP/Public/HTTPSVR/HTML using the defaults SUBTREE(*ALL) PATTERN(*.HTM*). The subdirectories will be searched and only files containing the pattern *.HTM will be included in the list.

Example 2: Create an index

```
CFGHTTPSCH OPTION(*CRTIDX) IDX(myindex)
DOCLIST('/QIBM/USERDATA/HTTPSVR/INDEX/+
myindex.DOCUMENT.LIST')
IDXHTML(*ABSTRACT)
```

This example will create an index called 'myindex' in index directory /QIBM/USERDATA/HTTPSVR/INDEX. The document list is in the file /QIBM/USERDATA/HTTPSVR/INDEX/myindex.DOCUMENT.LIST.

In this example the following is defined:

- The documents are HTML documents by default.
- Any file errors found for a document are ignored.
- Searches can be case-sensitive.
- Alphanumeric characters are valid search characters.
- The META tag with "Abstract" will be indexed separately.
- The character string following the META tag will also be included when the document is indexed.
- Searches are enabled for the entire document and the META tag field.

Example 3: Create a mapping rules file

```
CFGHTTPSCH OPTION(*CRTMAPF) CFG('MYCFG')
URLPFX('http://www.myserver.com')
MAPFILE(/QIBM/USERDATA/HTTPSVR/INDEX/myindex.MAP_FILE)
```

This example will create a mapping file called '/QIBM/USERDATA/HTTPSVR/INDEX/myindex.MAP_FILE'. The URL prefix 'http://www.myserver.com' plus all of the Pass directives from the MYCFG configuration will be copied to the mapping rules file. When documents are found on a search, the URLPFX will be followed by the path determined from the actual file path and the Pass directive.

If a document is physically located at '/root/clothing/doc1.htm', and there is a Pass /clothing/* /root/clothing/* directive in the configuration file, the URL for the document on the search results will be http://www.myserver.com/clothing/doc1.htm.

Error messages for CFGHTTPSCH

***ESCAPE Messages**

HTP1603

Request to add or delete documents was incomplete. Reason &1.

HTP1608

Request to create an index failed. Reason &1.

HTP1609

Request to add or delete documents from an index failed. Reason &1.

HTP160A

Request to merge an index failed. Reason &1.

HTP160B

Request to delete an index failed. Reason &1.

HTP160C

Request to create or append to a document list failed. Reason &1.

HTP160D

Request to delete a document list failed. Reason &1.

HTP160E

Request to create or append to a mapping rules file failed. Reason &1.

HTP160F

*IOSYSCFG special authority required to use CHGHTTPSCH command.

HTP1621

Request to create an index was incomplete. Reason &1.

HTP1623

Request to create a thesaurus dictionary failed. Reason &1.

HTP1624

Request to delete a thesaurus dictionary failed. Reason &1.

HTP1625

Request to retrieve a definition file failed. Reason &1.

HTP164B

Request to create a validation list failed. Reason &1.

HTP164C

Request to add or remove entries from a validation list failed. Reason &1.

HTP164D

Request to delete a validation list failed. Reason &1.

HTP164F

Request to create or update an options object failed. Reason &1.

HTP165A

Request to delete an options object failed. Reason &1.

HTP165B

Request to create or update a URL object failed. Reason &1.

HTP165C

Request to delete URL object failed. Reason &1.

HTP165F

Request to register a document list failed. Reason &1.

HTP1666

Request to create a validation list was incomplete. Reason &1.

HTP1668

Request to add or remove entries from a validation list was incomplete. Reason &1.

HTP1669

Request to create or update a URL object was incomplete. Reason &1.

HTP166C

Request to create or update a document list was incomplete. Reason &1.

HTP166D

Request to print the status of an index failed. Reason &1.

HTP166E

Request to print the status of a document list failed. Reason &1.

CFGPM400 (Configure PM/400) Command Description

CFGPM400 Command syntax diagram

Purpose

The Configure Performance Management/400 (CFGPM400) command allows you to set up your iSeries 400 to send and receive PM/400 performance data.

For more information about Performance Management/400, see these Web sites:

- iSeries Information Center at: <http://www.ibm.com/eserver/series/infocenter>
- Performance Management home page: <http://www.ibm.com/eserver/series/pm400>

There are no parameters for this command.

Example for CFGPM400

```
CFGPM400
```

This command displays a series of panels to activate and configure PM/400. This will enable your iSeries 400 performance data to be sent to IBM. You can also define the PM/400 contact information.

Error messages for CFGPM400

None

CFGTCPPTP (Configure Point-to-Point TCP/IP) Command Description

CFGTCPPTP Command syntax diagram

Purpose

The Configure Point-to-Point TCP/IP (CFGTCPPTP) command is used to display a menu that allows you to define, change, or display the TCP/IP point-to-point configuration.

There are no parameters for this command.

Example for CFGTCPPTP

```
CFGTCPPTP
```

This command displays the Configure Point-to-Point TCP/IP menu.

Error messages for CFGTCPPTP

***ESCAPE Messages**

TCP1A1F

Cannot process request while &3/&2/&1 using &6.

CFGSYSSEC (Configure System Security) Command Description

CFGSYSSEC Command syntax diagram

Purpose

The Configure System Security (CFGSYSSEC) command activates security features of the system by turning on security auditing, changing system values, and modifying system-supplied user profiles.

This command can be customized by the security administrator by following the steps below:

1. Issue the Retrieve CL Source (RTVCLSRC) command against the program QSECCFGS.
2. Edit the source code produced from the RTVCLSRC command and compile the new program. Make sure that the program is given a new name, is created into a library other than QSYS, and that the *PUBLIC authority is set to *EXCLUDE.
3. Issue the Change Command (CHGCMD) against the Configure System Security command and specify your new program for the **PGM** parameter. An example is listed below.

Note:

If a product upgrade is done, the CFGSYSSEC command is reinstalled, or if maintenance is applied to the CFGSYSSEC command, the CHGCMD will have to be issued again to customize the command.

There are no parameters for this command.

Restrictions: You must have *ALLOBJ, *SECADM, and *AUDIT special authorities to use this command.

Example for CFGSYSSEC

```
CFGSYSSEC
```

This command allows you to configure the security features of your system.

Error messages for CFGSYSSEC

***ESCAPE Messages**

CPFB304

User does not have required special authorities.

CFGTCP (Configure TCP/IP) Command Description

CFGTCP Command syntax diagram

Purpose

The Configure TCP/IP (CFGTCP) command is used to display a menu that allows a user to define or change the Transmission Control Protocol/Internet Protocol (TCP/IP) configuration. There are no parameters for this command.

Example for CFGTCP

```
CFGTCP
```

This command displays the Configure TCP/IP menu.

Error messages for CFGTCP

*ESCAPE Messages

TCP1D03

&1 member record length not correct.

TCP1D04

Error occurred processing member &1 of &2/&3.

TCP9999

Internal system error in program &1.

CFGTCPAPP (Configure TCP/IP Applications) Command Description

CFGTCPAPP Command syntax diagram

Purpose

The Configure TCP/IP Applications (CFGTCPAPP) command is used to define or change the application configuration for Transmission Control Protocol/Internet Protocol (TCP/IP).

Optional Parameter

APP Specifies the application being configured.

***SELECT:** Displays the Configure TCP/IP Applications selection menu. From this menu you can select which of the TCP/IP applications you want to configure. If the TCP/IP Licensed Program Product is installed, all application configuration options are displayed. Otherwise, only the SNMP configuration option is displayed.

***SNMP:** Configure the Simple Network Management Protocol (SNMP) Agent application. This directly calls the Configure TCP/IP SNMP (CFGTCPSNMP) command.

***ROUTED:** Configure the Routing Information Protocol (RouteD) server application. This directly calls the Configure TCP/IP Routed (CFGTCPRTD) command.

***TFTP:** Change Trivial FTP (TFTP) attributes. This directly calls the Change TFTP attributes (CHGTFTP) command.

***BOOTP:** Configure the boot protocol (BOOTP) server. This directly calls the Change BOOTP (CHGBPA) command or work with the BOOTP table.

***DDM:** Change DDM server attributes. This directly calls the Change DDM attributes (CHGDDMTCPA) command.

***DHCP:** Change the Dynamic Host Configuration Protocol (DHCP) attributes. This directly calls the Change DHCP attributes (CHGDHCPA) command.

***TELNET:** Configure the TELNET application. This directly calls the Configure TCP/IP TELNET (CFGTCPTELN) command. This value is valid only if the TCP/IP Licensed Program product has been installed.

***FTP:** Change File Transfer Protocol (FTP) attributes. This directly calls the Change FTP Attributes (CHGFTP) command. This value is valid only if the TCP/IP Licensed Program product has been installed.

***SMTP:** Configure the Simple Mail Transfer Protocol (SMTP) application. This directly calls the Configure TCP/IP SMTP (CFGTCPSMTP) command. This value is valid only if the TCP/IP Licensed Program product has been installed.

***LPD:** Change line printer daemon (LPD) attributes. This directly calls the Change LPD Attributes (CHGLPDA) command. This value is valid only if the TCP/IP Licensed Program product has been installed.

***HTTP:** Configure the World Wide Web HyperText Transfer Protocol (HTTP) server. This directly calls the Configure TCP/IP HTTP (CFGTCPHTTP) command.

***POP:** Configure the Post Office Protocol (POP) Version 3 mail server.

***REXEC:** Change TCP/IP Remote EXECution (REXEC) server attributes. This directly calls the Change REXEC Attributes (CHGRXCA) command. This value is valid only if the TCP/IP Licensed Program product has been installed.

***DNS:** Change the Dynamic Name Server (DNS) attributes. This directly calls the Change DNS attributes (CHGDNSA) command.

Examples for CFGTCPAPP

Example 1: Configuring TCP/IP Applications

```
CFGTCPAPP
```

This command displays the Configure TCP/IP Applications menu.

Example 2: Configuring TCP/IP TELNET Applications

```
CFGTCPAPP APP(*TELNET)
```

This command displays the Configure TCP/IP TELNET Applications menu.

Error messages for CFGTCPAPP

*ESCAPE Messages

TCP9999

Internal system error in program &1.

CFGTCPBP (Configure TCP/IP BOOTP) Command Description

CFGTCPBP Command syntax diagram

Purpose

The Configure TCP/IP BOOTP (CFGTCPBP) command allows you to work with bootstrap protocol (BOOTP) configuration commands.

There are no parameters for this command.

Example for CFGTCPBP

```
CFGTCPBP
```

This command displays the Configure TCP/IP BOOTP menu.

Error messages for CFGTCPBP

None

CFGTCPHTTP (Configure TCP/IP HTTP) Command Description

Note: To use this command, you must have the 5722-DG1 (HTTP Server) product installed.

Purpose

The Configure TCP/IP HTTP (CFGTCPHTTP) command shows a menu that allows you to set the World Wide Web Hypertext Transfer Protocol (HTTP) server configuration. This command shows the Configure TCP/IP HTTP menu. By using the CFGTCPHTTP command, you can change HTTP attributes or work with the HTTP configuration. Additionally, if you have installed IBM iSeries TCP/IP Utilities (57xx-TC1), then you can configure the IBM iSeries Workstation Gateway (WSG) by using this command.

Restriction: There are no restrictions for this command.

Parameters

There are no parameters for this command.

Error messages for CFGTCPHTTP

*ESCAPE messages

TCP8050

*IOSYSCFG authority required to use &1.

CFGTCPRTD (Configure TCP/IP Routed) Command Description

CFGTCPRTD Command syntax diagram

Purpose

The Configure TCP/IP Routed (CFGTCPRTD) command is used to display a menu that allows a user to define or change the Routed configuration.

There are no parameters for this command.

Restrictions:

You must have *IOSYSCFG special authority to use this command.

Example for CFGTCPRTD

```
CFGTCPRTD
```

This command shows the Configure TCP/IP Routed menu.

Error messages for CFGTCPRTD

None

CFGTCPSNMP (Configure TCP/IP SNMP) Command Description

CFGTCPSNMP Command syntax diagram

Purpose

The Configure TCP/IP SNMP (CFGTCPSNMP) command is used to display a menu that allows a user to define or change the Simple Network Management Protocol (SNMP) configuration. The menu options include:

- Change SNMP attributes
- Work with communities for SNMP

It is not necessary to run the CFGTCPSNMP command before using the SNMP agent. The SNMP agent is shipped with a community that has the following characteristics:

Community Name

public

ASCIICOM

*YES

INTNETADR

*ANY

OBJACC

*READ

LOGSET

*NO

LOGGET

*NO

See the Change SNMP Attributes (CHGSNMPA) command for the default values for SNMP attributes.

There are no parameters for this command.

Example for CFGTCPSNMP

```
CFGTCPSNMP
```

This command displays the Configure TCP/IP SNMP menu.

Error messages for CFGTCPSNMP

***ESCAPE Messages**

TCP4001

Error occurred accessing SNMP configuration information.

CVTCLSRC (Convert CL Source) Command Description

CVTCLSRC Command syntax diagram

Purpose

The Convert CL Source (CVTCLSRC) command is used to convert CL source from System/38 syntax to an iSeries 400 syntax.

The CVTCLSRC command converts the following to an iSeries 400 syntax:

- System/38 *object-name.library name* to iSeries 400 *library-name/object-name* qualification
- System/38 *job-name[.user-name(.job-number)]* to iSeries 400 *[(job-number)/user-name/]job-name* qualification
- Starting comment delimiters: *(/**) to *(/*)* when necessary
- Command names

- Keyword names and values
- Missing required parameters for iSeries 400 commands may need to be added.

Conversion of user-defined commands is limited to the reordering of qualified names and adjusting comment syntax.

The CVTCLSRC command creates a report indicating the success or failure of the source file conversion. This report is contained in a printer file with the name 'CVTCLSRC'. Successful conversions of System/38 source are noted in the report with the message:

```
CPF0786 Member has been converted.
```

Error messages are printed for unsuccessful conversions. Some examples of functions which cannot be converted and may be printed as error messages in the report are:

```
stmt# CPF0785 Command cannot be converted
stmt# CPF0789 Keyword cannot be converted
```

The user may write a program, perhaps by using the Copy Spooled File (CPYSPLF) command, to process the report based on the success or failure of the conversion.

Restriction: Library QSYS38 must exist on the system for conversion of command names, keywords, keyword values, and detection of unsupported functions by this command. Commands with unsupported command names, keyword names, or keyword values are not converted.

Required Parameters

FROMFILE

Specifies the qualified name of the System/38 source file to convert to iSeries 400 syntax.

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.

file-name: Specify the name of the System/38 source file to convert to iSeries 400 syntax.

TOFILE

Specifies the qualified name of the file in which the converted source is placed. It must be different than the name of the FROMFILE.

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.

file-name: Specify the name in which the converted source file is placed.

FROMMMBR

Specifies the member name of the source file member to convert.

***ALL**: All members of the specified source file are converted to iSeries 400 syntax.

generic-member-name*: Specify the generic name of the source file members to convert. 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. See generic names for additional information.

member-names: Specify the names of the source file members to convert. Specify no more than 50 names.

Note that the member name of the *converted* source member is the same as the member name of the unconverted source member in the file specified on the FROMFILE parameter.

Example for CVTCLSRC

```
CVTCLSRC FROMFILE(OLDLIB/FILEA) TOFILE(NEWLIB/FILEB)
FROMMMBR(PGM1 PGM2 PGM3)
```

This command converts three members (PGM1, PGM2, PGM3) of a System/38 source file (FILEA) located in library OLDLIB, to an iSeries 400 source file. The converted source file members are located in FILEB, in library NEWLIB. The converted members keep their original member names, PGM1, PGM2, and PGM3.

Error messages for CVTCLSRC

*ESCAPE Messages

CPF0781

File &1 in library &2 not a source file.

CPF0784

Specified to-file same as from-file.

CVTDAT (Convert Date) Command Description

CVTDAT Command syntax diagram

Purpose

The Convert Date (CVTDAT) command converts a date value from one format to another, without changing its value. The command ignores any date separators used in the old format, but if separators are desired in the converted result, a separator character can be specified on the command. Only valid dates can be converted. If either the from-format or the to-format use only 2 digits to specify the year (for example, *MDY, *DMY, *YMD, or *JUL), valid dates are in the range of January 1, 1940, to December 31, 2039. Otherwise, valid dates are in the range of August 24, 1928, to May 9, 2071. If the year is specified with only 2 digits, years in the range of 40 to 99 are assumed to be 1940 to 1999; years in the range 00 to 39 are assumed to be 2000 to 2039. The command works in conjunction with the QLEAPADJ system value.

Restriction: This command is valid only within a control language (CL) program.

Required Parameters

DATE Specifies the constant or CL variable that contains the date to be converted. When a constant is specified that contains separator characters, it must be enclosed in apostrophes (the separator

characters are ignored in the conversion). If separators are used in a constant, leading zeros in each part of the date can be omitted (3/3/88 or 03/03/88 are both valid). If a variable is specified, it must be long enough to contain the date type and its date separators, if used. The valid date separators are the slash (/), hyphen (-), period (.), and comma (,). A variable containing all blanks (X'40') is considered to have a date of length zero, and is not valid.

TOVAR

Specifies the name of the CL variable to receive the converted date value.

For every format except Julian, the month and day subfields in the converted result are each 2 bytes in length, are right-justified, and (if necessary) a leading zero is used as a padding character to fill each 2-byte field. The following table explains the field sizes and minimum variable lengths for the various formats.

For the Julian and long Julian formats, the day field is 3 bytes long and padded with leading zeros (if necessary). The year field is 2 bytes long for Julian and 4 bytes long for long Julian.

Use the following table to determine the required minimum length of the variable.

Table 1. Field Size and Minimum Variable Length

TO FMT	TO SEP	Minimum Variable length
*JUL	*NONE	5
*JUL	Any	6
*MDY, *DMY, *YMD	*NONE	6
*MDY, *DMY, *YMD	Any	8
*MDYY, *DMYY, *YYMD	*NONE	8
*MDYY, *DMYY, *YYMD	Any	10
*CYMD	*NONE	7
*CYMD	Any	9
*LONGJUL	*NONE	7
*LONGJUL	Any	8
*ISO, *USA, *EUR, *JIS		10
*JOB		Depends on job date format
*SYSVAL		Depends on value of QDATFMT

TO FMT	Month	Field Size	
		Day	Year
*JUL	N/A	3	2
*MDY, *DMY, *YMD	2	2	2
*MDYY, *DMYY, *YYMD	2	2	4
*ISO, *USA, *EUR, *JIS	2	2	4
*CYMD	2	2	2 (+1 byte century field)

Optional Parameters

FROMFMT

Specifies the current format of the date being converted.

***JOB:** The format specified on the job attribute DATFMT is used.

***SYSVAL:** The format specified by the system value QDATFMT is used.

***MDY:** The date has the month, day, year format, **mmddyy**.

***MDYY:** The date has the month, day, year format, **mmddyyyy**.

- ***DMY**: The date has the day, month, year format, **ddmmyy**.
- ***DMYY**: The date has the day, month, year format, **ddmmyyyy**.
- ***YMD**: The date has the year, month, day format, **yymmdd**.
- ***YYMD**: The date has the year, month, day format, **yyymmdd**.
- ***CYMD**: The date has the century, year, month, day format, **cyymmdd**, where c is 0 for years 1928 through 1999 and is 1 for years 2000 through 2071.
- ***JUL**: The date has the Julian format, **yyddd**.
- ***ISO**: The date has the International Organization for Standardization (ISO) date format, **yyyy-mm-dd**.
- ***USA**: The date has the United States date format, **mm/dd/yyyy**.
- ***EUR**: The date has the European date format, **dd.mm.yyyy**.
- ***JIS**: The date has the Japanese Industrial Standard date format, **yyyy-mm-dd**.
- ***LONGJUL**: The date has the long Julian format, **yyyyddd**.

TOFMT

Specifies the format to which the date is being converted.

- ***JOB**: The format specified on the job attribute DATFMT is used.
- ***SYSVAL**: The date is converted to the format specified by the system value QDATFMT.
- ***MDY**: The date format is converted to the month, day, year format, **mmddy**.
- ***MDYY**: The date format is converted to the month, day, year format, **mmddyyy**.
- ***DMY**: The date format is converted to the day, month, year format, **ddmmy**.
- ***DMYY**: The date format is converted to the day, month, year format, **ddmmyyy**.
- ***YMD**: The date format is converted to the year, month, day format, **yymmdd**.
- ***YYMD**: The date format is converted to the year, month, day format, **yyymmdd**.
- ***CYMD**: The date format is converted to the century, year, month, day format, **cyymmdd**, where c is 0 for years 1928 through 1999 and is 1 for years 2000 through 2071. If the year in the current format is only 2 digits, c will be set to 0 for years 40 through 99 and to 1 for years 00 through 39.
- ***JUL**: The date format is converted to the Julian format, **yyddd**.
- ***ISO**: The date format is converted to the International Organization for Standardization (ISO) date format, **yyyy-mm-dd**.
- ***USA**: The date format is converted to the United States date format, **mm/dd/yyyy**.
- ***EUR**: The date format is converted to the European date format, **dd.mm.yyyy**.
- ***JIS**: The date format is converted to the Japanese Industrial Standard date format, **yyyy-mm-dd**.
- ***LONGJUL**: The date has the long Julian format, **yyyyddd**.

TOSEP

Specifies the type of date separators, if any, used in the converted date.

- ***JOB**: The converted date has the separators specified by the job attribute DATSEP.
- ***SYSVAL**: The converted date has the separators specified by the system value QDATSEP.
- ***NONE**: No separator characters are contained in the converted date.
- ***BLANK**: A blank is used as the separator in the converted date.

separator-character: Specify the character that is used as the date separator in the converted date. The valid separator characters are the slash (/), hyphen (-), period (.), and comma (,).

Examples for CVTDAT

Example 1: Converting to DMY Format

```
DCL  VAR(&DATE)  TYPE(*CHAR)  LEN(8)
*
*
*
CVTDAT  DATE('12-24-88')  TOVAR(&DATE)  TOFMT(*DMY)
```

This command converts the date 12-24-88, which is in the MDY format. Because the FROMFMT parameter was not specified, its default, *JOB, indicates that the job attribute DATFMT contains the MDY format. The date is converted to the DMY format, and the separator character specified in the job attribute DATSEP is inserted. If DATSEP contains a slash, the converted result is 24/12/88.

Example 2: Converting to Format Specified by Job Attribute

```
DCL  &PAYDAY  *CHAR  6
DCL  &NEWPDAY *CHAR  6
*
*
*
CVTDAT  DATE(&PAYDAY)  TOVAR(&NEWPDAY)
        FROMFMT(*YMD)  TOSEP(*NONE)
```

This command converts the format of the date stored in &PAYDAY from year, month, day to the format specified by the job attribute DATFMT. If, for example, DATFMT contains the MDY format, the format of the converted date is month, day, and year. The converted date is stored in the variable &NEWPDAY. Because &NEWPDAY was declared as a 6-character variable, TOSEP(*NONE) is required; the converted result cannot include separator characters.

Example 3: Converting to CYMD format

```
DCL  &NEWDAY1 *CHAR  7
DCL  &NEWDAY2 *CHAR  7
*
*
*
CVTDAT  DATE('01/24/1939') TOVAR(&NEWDAY1)
        FROMFMT(*MDYY) TOFMT(*CYMD) TOSEP(*NONE)
CVTDAT  DATE('01/24/39') TOVAR(&NEWDAY2)
        FROMFMT(*MDY)  TOFMT(*CYMD) TOSEP(*NONE)
```

The first CVTDAT command converts the date specified on the DATE parameter from the month, day, 4-digit year format to the century, year, month, day format. Because the year was specified with 4 digits and the first 2 digits are "19", the century digit is set to "0", so the value of "NEWDAY1 is "0390124".

The second CVTDAT command converts the date specified on the DATE parameter from the month, day, year format to the century, year, month, day format. Because the year was specified with only 2 digits and the year is less than 40, the century digit is set to "1", so the value of "NEWDAY2 is "1390124".

Error messages for CVTDAT

*ESCAPE Messages

CPF0550

Date too short for specified format.

CPF0551

Separators in date are not valid.

CPF0552

Date contains misplaced or extra separators.

CPF0553

Date contains too many or too few numeric characters.

CPF0554

Variable specified too short for converted date format.

CPF0555

Date not in specified format or date not valid.

CPF0556

Date contains two or more kinds of separators.

CPF0557

Date outside allowed range.



CVTDIR (Convert Directory) Command Description

CVTDIR Command syntax diagram

Purpose

The Convert Directory (CVTDIR) command can provide information on converting integrated file system directories from *TYPE1 format to *TYPE2 format, or perform a conversion. *TYPE2 directories are optimized for performance, size, and reliability as compared to directories having the *TYPE1 format. The information provided includes estimates of the amount of time a conversion will take, the current directory format of the file systems and disk storage requirements of the conversion.

Restrictions:

1. Only directories in root (/) and QOpenSys file systems, and in basic User ASPs can be converted or estimated.
2. The system must be in a restricted state if the directories in the root (/) or QOpenSys file systems are being converted.
3. You must have *ALLOBJ authority to use this command.
4. When you have specified OPTION(*CONVERT), the job cannot be canceled. If the system abnormally ends while processing the conversion, the conversion will complete during the ensuing IPL.

Required Parameter

OPTION

Specifies the function you want to perform.

***CHECK:** The file systems which are currently on the system are checked to determine whether they are eligible for conversion. Message CPIA084 is sent for the root (/), and QOpenSys file systems, and for all active auxiliary storage pools on the system identifying their current directory format.

***ESTIMATE:** An estimate is provided of the time it will take to convert the specified file systems to the specified directory format. Information is also provided on disk storage considerations for the conversion. The information is returned in messages CPIA087, CPIA090 and CPIA091. Additionally, the system will pre-create objects which will reduce the time it takes to do a subsequent conversion.

Note: We recommend that you run the Reclaim Storage (RCLSTG) command prior to using this option, if a RCLSTG has not recently been done.

***CONVERT:** All of the directories in the specified file system are converted to the specified directory format. This may be a long running function. Therefore, it is strongly recommended that you first use the *ESTIMATE option to see how long this processing will take, and to pre-create some objects that can be used by the directory conversion.

Additional disk storage will be required when running the conversion. The amount of storage required will be proportional to the depth of the directories and the number of disk units in the ASP.

Additionally, this option will rename any object names that are not valid for UTF-16. Message CPIA08A will be sent for each object name that is renamed. See Integrated file system topic in the File systems and management category of the Information Center for more information.

Note: We recommend that you run the Reclaim Storage (RCLSTG) command prior to using this option, if a RCLSTG has not recently been done.

Note: When the root (/) and QOpenSys file systems are converted, all user-defined file systems will be unmounted by the system. When the user-defined file systems in a specific ASP are converted, all user-defined file systems in that ASP will be unmounted. These unmounted file systems will not be remounted by the system when the conversion is complete.

Optional Parameters

FILESYS

Specifies the file system whose directories you want to process.

***NONE:** No file system will be processed.

***ROOT:** The directories in the root (/) file system will be processed.

***QOPENSYS:** The directories in the QOpenSys file system will be processed.

***UDFS:** The directories in the user-defined file systems on the specified auxiliary storage pool (ASP parameter) will be processed.

Note: If FORMAT(*TYPE1) is specified, then the specified ASP must have no user-defined file systems.

***ALL:** All of the directories in the eligible file systems will be processed. If OPTION(*CONVERT) is specified, inquiry message CPAA084 will be sent listing the file systems which will be converted and requesting confirmation of the conversion.

ASP Specifies the auxiliary storage pool (ASP) number of the user-defined file systems you want to process.

ASP-number: Specify a value ranging from 1 through 32 to specify the number of the ASP whose user-defined file systems you want to process. Valid values depend on how ASPs are defined on the system.

Note:

The value of 1 is the system ASP, any other value indicates a user ASP. If the system ASP is specified, only the user-defined file systems in the system ASP are converted; that is, root (/) and QOpenSys will not be converted.

FORMAT

Specifies the directory format type you want to convert to.

***TYPE2:** The directories in the specified file system will be converted to *TYPE2. Any existing directories will be converted to *TYPE2 and any new directories created in the specified file system after this conversion will be *TYPE2. This directory format is optimized for performance, size, and reliability as compared to directories having the *TYPE1 format.

***TYPE1:** The specified file system will be converted so that any new directories created in the specified file system after this conversion will be *TYPE1. This is the original directory format which was available when the integrated file system was first introduced.

DETAIL

Specifies the amount of information which should be periodically displayed in informational status messages while the directories are being converted.

***BASIC:** The number of links which have been processed will be displayed. Choosing this option will reduce the time it takes to do the directory conversion.

***EXTENDED:** The percentage of the total number of links which have been processed will be displayed as well as the estimated number of minutes remaining for the conversion process to complete. Choosing this option will increase the time it takes to do the directory conversion because the entire directory tree will be scanned twice. The amount of additional time will be proportional to the number of directories which must be processed.

Examples for CVTDIR**Example 1: Checking Directory Format Information**

```
CVTDIR OPTION(*CHECK)
```

This command checks which file systems can be converted and returns the current directory formats for the file systems.

Example 2: Converting Directories in a User ASP to *TYPE2 format

```
CVTDIR OPTION(*CONVERT) FILESYS(*UDFS) ASP(8) FORMAT(*TYPE2)
```

This command converts the directories in the user-defined file systems on auxiliary storage pool 8 to *TYPE2 format.

Error messages for CVTDIR***ESCAPE Messages****CPFA099**

The requested directory conversion cannot be performed.

CPFA09A

Errors occurred during directory conversion.



CVTDLSNAM (Convert Document Library Services Name) Command Description

CVTDLSNAM Command syntax diagram

Purpose

The Convert Document Library Services Name (CVTDLSNAM) command is used before or after a change in the CCSID assumed for EBCDIC object names by QDLS (the document library services file system). The command can help identify QDLS object names that may be different because of the CCSID change, and it can rename QDLS objects so they retain their original names. It can also correct the names of objects whose names changed when they were copied between QDLS and another file system. The command prints a report with the job's spooled output that lists selected objects and any actions taken on those objects.

When converting object names to or from EBCDIC, QDLS uses the job default CCSID unless data area QUSRSYS/QODEC500 exists, in which case QDLS uses CCSID 500 (the data area allows reversion to the behavior of early versions of QDLS). The CCSID used by QDLS is therefore changed by creating or deleting the data area, or by changing the job default CCSID when the data area does not exist.

The CCSID affects the view of QDLS object names by integrated file system clients of QDLS, which must convert object names to and from EBCDIC. Those clients include:

- integrated file system commands such as DSPLNK, CPY, MOV, and RNM
- UNIX-type APIs provided by the integrated file system, such as access, open, rename, and unlink
- IBM Client Access Windows Client for OS/400 Version 3
- IBM Client Access Optimized OS/2 Client for OS/400 Version 3

The CCSID does not affect clients of QDLS that work directly with EBCDIC object names, which include:

- document and folder commands, such as CRTDOC, CPYDOC, WRKDOC, CRTFLR, WRKFLR, DLTDL0, and RNMDLO
- hierarchical file system (HFS) APIs, such as QHFDLTSF, QHFOPNDR, QHFOPNSF, and QHFRNMSF
- IBM Client Access OS/2 Client for OS/400 Version 3
- IBM Client Access DOS Client for OS/400 Version 3
- IBM Client Access DOS with Extended Memory Client for OS/400 version 3

Even for integrated file system clients of QDLS, the CCSID doesn't matter except for objects that are also used by EBCDIC clients. In that case, QDLS object names may appear different to the clients if the names contain variant characters and the clients are using different CCSIDs (integrated file system clients use the CCSID as described earlier, and EBCDIC clients likely use the job default CCSID).

Restrictions:

1. You must have *R authority to the directory containing the object links and *X to the other directories in the path.
2. The additional authority restrictions from the RNM command apply when renaming objects.

Required Parameter

OBJ Specifies the objects to process. A maximum of 300 path names can be specified; however, all paths must be for the same file system. Each path name can be either a simple name or a name that is qualified with the name of the directory in which the object is located. A pattern can be specified in the last part of the path name: an asterisk (*) matches any number of characters and a question mark (?) matches any single character. If a path name is qualified or contains a pattern, it must be enclosed in apostrophes (').

For more information on specifying path names, refer to path names and the Integrated File System Introduction topic in the Information Center.

The CVTDLSNAM command is an integrated file system client of QDLS. As such, its view of object names can be different than that of EBCDIC clients. So, use care in specifying object names. It is generally safer to use generic characters in place of variant characters (for example, specify X?X as an object name rather than X!X).

Optional Parameters

SUBTREE

Specifies whether directory subtrees are processed.

***OBJ:** Only the objects that match the given path names are processed. If a path name specifies a directory, objects in the directory are not processed.

***DIR:** Objects in the first level of each directory that matches a given path name are processed.

***ALL:** The entire subtree of each directory that matches a given path name is processed.

ACTION

Specifies the action to perform on the selected objects.

***LIST:** For QDLS, this value lists the selected objects that might appear to have different names if the CCSID assumed by QDLS for EBCDIC object names is changed from the specified old value to the specified new value. For other file systems, this value lists the selected objects that might have an unexpected name after having been copied from QDLS, and neither specified CCSID is used in this case.

***RENAME:** Corrects the names of the selected objects. If *RENAME is used more than once on an object, the results will probably not be meaningful.

Some objects may fail to be renamed when requested, such as if the new name already exists. However, the command will not fail immediately; it will continue to process any remaining objects.

For QDLS, *RENAME will change the names such that, after changing the CCSID assumed by QDLS for EBCDIC object names from the specified old value to the specified new value, the object names will appear the same as before the change to integrated file system clients of QDLS.

For other file systems, the specified objects are presumed to have been created with the specified old CCSID and implicitly renamed as they were copied from QDLS by an integrated file system client of QDLS using the specified new CCSID. *RENAME will change the names of the objects to be the same as those of the original QDLS objects.

Note: The effect of a rename can be undone by another rename with the CCSIDs reversed. For example, if a rename is done using FROMCCSID(500) and TOCCSID(273), the original name(s) can be restored by a rename using FROMCCSID(273) and TOCCSID(500).

PREVIEW

Selects whether to preview the results of the selected action.

***NO:** Perform the selected action.

***YES:** Inhibit the selected action and report what the results would be. This value is allowed only when ACTION(*RENAME) is specified.

FROMCCSID

Specifies the original coded character set identifier (CCSID) of the EBCDIC object name. This value is ignored when processing objects in file systems other than QDLS if ACTION is *LIST.

500: CCSID 500 is used. That is the CCSID used by early versions of QDLS.

***JOB:** The current job's default CCSID is used.

***SYSVAL:** The CCSID specified in the system value QCCSID is used.

***HEX or 65535:** The CCSID currently assumed by QDLS for EBCDIC object names is used.

from-coded-character-set-identifier: Specify the CCSID to be used. More information on valid CCSIDs is in the Globalization topic in the Information Center.

TOCCSID

Specifies the coded character set identifier (CCSID) assumed by QDLS for EBCDIC object names. This value is ignored when processing objects in file systems other than QDLS if ACTION is *LIST.

***JOB:** The current job's default CCSID is used.

***SYSVAL:** The CCSID specified in the system value QCCSID is used.

***HEX or 65535:** The new CCSID is unknown, such as when different jobs will be using different CCSIDs. This value may not be used with ACTION(*RENAME).

to-coded-character-set-identifier: Specify the CCSID to be used. More information on valid CCSIDs is in the Globalization topic in the Information Center.

Examples for CVTDLSNAM

Example 1: List QDLS Objects Affected by a CCSID Change

A new release of Operating System/400 is installed and QDLS now assumes the job default CCSID instead of CCSID 500 for EBCDIC object names. The job CCSID is currently set to 37. The following command is used to identify the objects that effectively have new names for integrated file system clients of QDLS. Note that the ACTION, FROMCCSID, and TOCCSID parameters could all have been omitted from the command, since they specify the default values in this case.

```
CVTDLSNAM OBJ('/QDLS') SUBTREE(*ALL)
ACTION(*LIST) FROMCCSID(500) TOCCSID(37)
```

Output similar to this might be produced:

```
/QDLS/FLRA/X] --> X!
/QDLS/FLRB/X! --> X]
```

Each line shows two names for an object, as it would be seen by clients using CCSID 500 and CCSID 37 (the second name won't be shown if TOCCSID is *HEX). The output shows that two objects are affected by the change of the assumed CCSID. The object known before the change as X] by integrated file system clients is known as X! afterward, and X! is renamed to X|.

The name X! seems more reasonable than either X] or X|, so we assume X! is the correct name in both cases. In the first case the new name is desirable; we surmise the object was created as X! by a client using CCSID 37. In the second case the new name is undesirable; the object was presumably created by a client using CCSID 500.

Example 2 - Rename QDLS Object to Adjust for a CCSID Change

The second object name from the example above is corrected using the following command. For this example the job CCSID is 500 (necessary to guarantee correct recognition of the object name X!). It is likely that a generic name (such as * or X? instead of X!) would be used in similar situations, eliminating the need to adjust the job CCSID.

```
CVTDLSNAM OBJ('/QDLS/FLRB/X!') ACTION(*RENAME)
FROMCCSID(500) TOCCSID(37)
```

This output might be produced:

```
/QDLS/FLRB/X! --> X]
```

Each line again shows two names for an object, but this time both names are what a CCSID 500 client would see. The output indicates that X! is renamed to X]. The new name may appear incorrect to a CCSID 500 client (X]), but it will appear as desired to a CCSID 37 client (X!).

Example 3 - Correct Names of Objects Copied from QDLS

A job with CCSID 273 uses the CRTDOC command to create several documents in folder FLR. The CPY command is used to copy all the objects from QDLS directory FLR to QLANSrv directory DIR when QDLS is using CCSID 500; some of the target objects are created with unexpected names. This command is used to correct those names:

```
CVTDLSNAM OBJ('/QLANSrv/DIR/*') ACTION(*RENAME)
FROMCCSID(273) TOCCSID(500)
```

This output might be produced:

```
/QLANSrv/DIR/X{ --> X ä
```

The output shows that one object, x{, was renamed, to X ä, its original QDLS name.

Error messages for CVTDLSNAM

*ESCAPE Messages

CPI8A22

Processing &1.

CVTEDU (Convert Education) Command Description

CVTEDU Command syntax diagram

Purpose

The Convert Education (CVTEDU) command converts course modules from ASCII to EBCDIC.

Note: Courses that are valid only for personal computers cannot be converted with this command.

Required Parameters

COURSE

Specifies the name of the course library to be converted.

***ALLADDED:** All of the courses previously added to the system through the education administration system will be converted.

course-ID: Specify the name of the course library to be converted.

Optional Parameters

LNG Specifies the language (indicated by the language identifier) of the course to be converted.

***SYSVAL:** The identifier of the current primary language for the system will be used.

language-ID: Specify the identifier of the language.

Example for CVTEDU

```
CVTEDU COURSE(*ALLADDED)
```

This command converts all of the courses previously added through the education administration system from ASCII to EBCDIC.

Error messages for CVTEDU

*ESCAPE Messages

CPF1D47

Not authorized to use CVTEDU command.

CPF1D49

Errors occurred during command processing.

CVTIPSIFC (Convert IP over SNA Interface) Command Description

CVTIPSIFC Command syntax diagram

Purpose

The Convert IP over SNA Interface (CVTIPSIFC) command converts an IP (Internet Protocol) address into its associated SNA network identifier and location name. The location entries defined with the Add IP over SNA Location Entry (ADDIPSLOC) CL command are searched to find the SNA location name and SNA network identifier associated with the input internet address (INTNETADR).

Required Parameter

INTNETADR

Specifies the internet address of the local host or a remote host to be converted.

The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Restrictions:

1. The internet address cannot begin with 0 (for example, 0.nnn.nnn.nnn).
2. The internet address cannot begin with 127 (for example, 127.nnn.nnn.nnn). This address range is reserved for TCP/IP loopback addresses.
3. The internet address cannot be a class D or class E address. Class D addresses range from 224.nnn.nnn.nnn to 239.nnn.nnn.nnn. Class E addresses range from 240.nnn.nnn.nnn to 255.nnn.nnn.nnn.

Optional Parameter

OUTPUT

Specifies where the result should be returned.

***:** The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is printed with the job's spooled output.

Examples for CVTIPSIFC

Example 1: Printing a Converted IP Address

```
CVTIPSIFC INTNETADR('128.1.2.3') OUTPUT(*PRINT)
```

This command finds the SNA network identifier and location name associated with IP address 128.1.2.3 and puts the result in the job's spooled output.

Example 2: Displaying a Converted IP Address

```
CVTIPSIFC  INTNETADR(128.2.3.4)
```

This command finds the SNA Network Identifier and Location Name associated with IP address 128.2.3.4 and puts the result to the display for an interactive job.

Error messages for CVTIPSIFC

*ESCAPE Messages

CPFA111

Internet address not converted.

CPFA118

No associated SNA network identifier and location name found.

CVTIPSLOC (Convert IP over SNA Location Entry) Command Description

CVTIPSLOC Command syntax diagram

Purpose

The Convert IP over SNA Location Entry (CVTIPSLOC) command is used to convert a specified SNA network identifier and location name into one or more associated IP addresses. The location entries defined with the ADD IP over SNA Location Entry (ADDIPSLOC) CL command are searched to find one or more IP addresses that are associated with the input SNA location name (LOC) and SNA network identifier (NETID).

Required Parameter

LOC Specifies the SNA location name to be converted.

location-name: Specifies the SNA location name for the local host or a remote host. This name can be one to eight characters in length. The first character must be A (or a) through Z (or z), or special characters \$, #, or @ followed by 0 through 9, A (or a) through Z (or z), \$, #, or @.

Optional Parameters

NETID Specifies the SNA network identifier for the local host or a remote host.

***NETATR**: Specifies that the network identifier in the network attributes for this host is used.

network-identifier: Specifies the network identifier for the local host or a remote host. This name can be one to eight characters in length. The first character must be A (or a) through Z (or z), or special characters \$, #, or @ followed by 0 through 9, A (or a) through Z (or z), \$, #, or @.

OUTPUT

Specifies where the results are returned.

*****: The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT**: The output is printed with the job's spooled output.

Example for CVTIPSLOC

```
CVTIPSLOC  LOC(LUNAMEX)  OUTPUT(*PRINT)
```

This command converts the location name LUNAMEX with the default network identifier specified in the network attributes and places the results in the job's spooled output.

Error messages for CVTIPSLOC

*ESCAPE Messages

CPFA115

SNA network identifier and location name not converted.

CPFA119

No associated internet address found.

CVTOPTBKU (Convert Optical Backup) Command Description

CVTOPTBKU Command syntax diagram

Purpose

The Convert Optical Backup (CVTOPTBKU) command converts an optical backup volume to an optical primary volume. User applications and programs can then write to the converted volume.

Note:

Once an optical volume is converted from a backup volume to a primary volume, you must initialize the optical volume to convert it to a backup volume again. Initializing an optical volume results in losing all existing information on the optical volume.

Restriction: To use this command you must have *ALL authority to the authorization list securing the volume to be converted.

Required Parameter

BKUVOL

Specifies the volume identifier of the optical backup volume being converted to a primary volume.


Optional Parameter

PRIVOL

Specifies the identifier of the optical volume after it is converted to a primary volume.

Note:

The identifier must be unique within the iSeries 400 optical system you are using. More information about optical

volume names can be found in the Optical Support  book.

***PRVPRIVOL:** The identifier of the new primary optical volume is the same as the identifier of the primary optical volume for which this volume previously was a backup.

When an optical backup volume is first used, the system records the volume identifier of the primary volume on the media. This is done to ensure that no other primary volume can use the same backup volume identifier. This also ensures that the original name of the primary volume is known at the time the optical backup volume is converted.

primary-volume-identifier: Specify a new volume identifier. This is the identifier of the volume after this command completes successfully.

Example for CVTOPTBKU

```
CVTOPTBKU BKUVOL(VOL01BACKUP) PRIVOL(VOL02)
```

This command converts the optical backup volume VOL01BACKUP to a primary optical volume. VOL02 is the identifier of the optical volume after it is converted.

Error messages for CVTOPTBKU

***ESCAPE Messages**

OPT1305

Optical volume &1 is read only.

OPT1315

Optical volume &1 is write protected.

OPT1320

Optical volume &1 in use.

OPT1325

Optical volume format not recognized.

OPT1330

Optical volume not found or not useable.

OPT1331

Optical volume &1 not found.

OPT1340

Optical volume &1 not initialized.

OPT1342

Invalid volume identifier specified.

OPT1345

No free space available on media.

OPT1350

Write operations failed to optical volume &1.

OPT1360

Media directory corrupted on optical volume &1.

OPT1375

Optical volume &1 already exists.

OPT1460

Optical volume &1 is not in an optical device.

OPT1462

Operation not completed, optical volume is not a backup volume.

OPT1530

&1 does not represent a valid optical device.

OPT1605

Media or device error occurred.

OPT1790

Operation conflicts with another request.

- OPT1805**
Error accessing optical volume index file.
- OPT1810**
Error accessing optical directory index file.
- OPT1815**
Internal program error occurred.
- OPT1820**
Internal error occurred on optical device &1.
- OPT1825**
Optical indexes are incorrect for optical device &1.
- OPT1860**
Request to optical device &1 failed.
- OPT1861**
No device description configured for resource &1.
- OPT1862**
No active device description for resource &1.
- OPT1863**
Optical libraries need to be reclaimed.
- OPT1872**
Optical request timed out.
- OPT2030**
Error during Convert Optical Backup.
- OPT2301**
Internal system object in use.
- OPT7740**
User not authorized to object &2 in library &3 type &4.

CVTPFRDTA (Convert Performance Data) Command Description

CVTPFRDTA Command syntax diagram

Purpose

The Convert Performance Data (CVTPFRDTA) command converts performance data from a previous release to the format needed for processing by the current release of Performance Tools for iSeries licensed program.

The command first determines the release level at which the data was collected. Then the members of all necessary files are converted. The conversion may be done in the same library where the current data resides. To avoid the risk of destroying the old data if the command ends abnormally, convert the data into a different library (TOLIB), and later, delete the data from the old library (FROMLIB).

If the conversion is done in a different library, the old data remains in the current library (FROMLIB) and the new data resides in the new library (TOLIB). If a new library is specified for the newly converted data, all files are copied to the new library, including those files which do not need to be converted.

Required Parameters

FROMLIB

Specifies the library that contains the files being converted.

TOLIB Specifies the library where the converted files will be located.

Optional Parameters

JOB Specifies the job description used to submit jobs for batch processing.

***USRPRF:** The job description specified in the user's user profile is used.

***NONE:** A batch job is not submitted. Instead, processing continues interactively while the user waits. The user's workstation is not available for other use during this time, which could be significant for long jobs.

The name of the job description 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.

job-description-name: Specify the name of an alternate job description.

Examples for CVTPFRDTA

Example 1: Submitting a Batch Job

```
CVTPFRDTA FROMLIB(MIKE) TOLIB(TERESA)
```

This command converts the performance data in library MIKE and places it in library TERESA.

Example 2: Submitting a Job Interactively

```
CVTPFRDTA FROMLIB(QPFRDATA)  
TOLIB(QPFRDATA) JOBD(*NONE)
```

This command converts the performance data in library QPFRDATA and places it in the same library after conversion is complete. This conversion occurs interactively while the user waits.

Error messages for CVTPFRDTA

*ESCAPE Messages

CPF0A0B

Performance Tools files did not convert.

CPF22F7

Number of authorities must be between 1 and &1.

CPF22FA

Authority value &1 not valid.

CPF22FB

Must specify *EXCLUDE or *AUTL as only authority value.

CPF2817

Copy command ended because of error.

CPF4102

File &2 in library &3 with member &4 not found.

CPF8122

&8 damage on library &4.

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9803

Cannot allocate object &2 in library &3.

CPF9807

One or more libraries in library list deleted.

CPF9808

Cannot allocate one or more libraries on library list.

CPF9810

Library &1 not found.

CPF9811

Program &1 in library &2 not found.

CPF9812

File &1 in library &2 not found.

CPF9820

Not authorized to use library &1.

CPF9830

Cannot assign library &1.

CVTPFRTHD (Convert Performance Thread Data) Command Description

CVTPFRTHD Command syntax diagram

Purpose

The Convert Performance Thread Data (CVTPFRTHD) command converts performance data records collected by Collection Services or data generated by the Create Performance Data (CRTPFRDTA) command. The specified member (MBR parameter) of database file QAPMJOBS contains records with thread-level performance data. You can use CVTPFRTHD to convert this data and write the resulting records to a member by the same name (MBR parameter) in file QAPMTJOB. The output file member will contain records with job-level performance data which are a total of the performance information for all threads running within the job.

The input file (QAPMJOBS or QAPMJOBL) must exist in the library specified on the LIB parameter. If file QAPMTJOB does not exist in the specified library (LIB parameter), it will be created automatically. A file member by the name specified (MBR parameter) will be automatically added to file QAPMTJOB if it did not already exist.

Required Parameters

MBR Specifies the member of QAPMJOBS file or QAPMJOBL file that contains the collections to be processed. This member will be created, if it does not exist already, or replaced in QAPMTJOB file.

member-name: Specify the name of the member containing thread-level performance data.

LIB Specifies the name of the library where file QAPMJOBS or file QAPMJOB1 resides, and where the QAPMTJOB file either resides or will be created.

QPRFDATA: The IBM-supplied performance data library, QPRFDATA, is used to locate database files.

library-name: Specify the library name where the database files are located.

Optional Parameter

REPLACE

Indicates whether the specified member in file QAPMTJOB will be replaced.

***YES**: If the member did not exist before, it is created. If the member already exists, the data contained in it is replaced.

***NO**: If the member did not exist before, it is created. If the member already exists, the data contained in it is not replaced and an error message is signalled.

Example for CVTPFRTHD

Example 1: Converting Performance Thread Data

CVTPFRTHD

This command converts performance data records.

Error messages for CVTPFRTHD

*ESCAPE Messages

CPF0A83

Performance thread data not converted.

CPF0A84

Member already exists.

CPF0A85

User profile &1 is not authorized to library &2.

CPF2110

Library &1 not found.

CPF2817

Copy command ended because of error.

CPF5030

Partial damage on member &4.

CPF9810

Library &1not found.

CPF9812

File &1in library &2 not found.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CVTRPCSRC (Convert RPC Source) Command Description

CVTRPCSRC Command syntax diagram

Purpose

The Convert RPC Source (CVTRPCSRC) command generates C code from an input file written in the Remote Procedure Call (RPC) Language. The generated C code can be used to implement an RPC protocol.

This command is equivalent to running the **rpcgen** utility on a UNIX system.

Required Parameter

FROMFILE

Specifies the path name of the input source file written in the Remote Procedure Call (RPC) Language. The input source file must be a file in the “root” (/) or QOpenSys file system.

For more information on specifying path names, refer to path names.

Optional Parameters

OPTION

Specifies the compile options.

***NOSAMP:** All file types except the sample files (*CLTSAMP and *SVRSAMP) are generated.

***ALL:** All file types are generated.

***XDR:** The input file is compiled into XDR (eXternal Data Representation) routines.

***HDR:** The input file is compiled into C data-definitions (a header file).

***CLTSTUB:** The input file is compiled into client-side stub procedures.

***SVRSTUB:** The input file is compiled into server-side stub procedures. However, no “main” routine is generated.

***CLTSAMP:** Sample client code that uses remote procedure calls is generated. The file can be customized for the application.

***SVRSAMP:** Sample server code that uses remote procedure calls is generated. The file can be customized for the application.

TOFILE

Specifies the path name of the output file. This option is only allowed if OPTION(*ALL) or OPTION(*NOSAMP) is not specified. When OPTION(*ALL) or OPTION(*NOSAMP) is specified, or if you do not specify the TOFILE parameter when using another option, the FROMFILE parameter is used to generate the TOFILE name as follows, where

`filename`

is the name of the input file name from the FROMFILE parameter.

- `filename.h` for a header file
- `filename_xdr.c` for an XDR file
- `filename_clnt.c` for client-side stubs
- `filename_svc.c` for server-side stubs
- `filename_client.c` for client-side sample files
- `filename_server.c` for server-side sample files

The output file or files for sample code must not exist; if any of the sample output files exist, the command will fail. Other output files will be overwritten if they exist.

'to-file-path name':

Specify a path name to be used to generate the TOFILE name or names.

For more information on specifying path names, refer to path names.

PROTOCOL

Compiles into server-side stub procedures for the transport that is specified. The specified value must be present in the /etc/netconfig file at the time the server application is run. This parameter is only used when OPTION(*SVRSTUB) is specified. One or more of the following options may be specified:

***NONE:** Compile server-side stub procedures for all transports that are in the /etc/netconfig file.

***TCP:** Compile server-side stub procedures for the TCP transport.

***UDP:** Compile server-side stubs for the UDP transport.

Examples for CVTRPCSRC

Example 1: Convert RPC Source to Default Files

```
CVTRPCSRC FROMFILE('/myrpc') OPTION(*ALL)
```

This converts the RPC language file '/myrpc' into all four file types, *XDR, *HDR, *CLTSTUB and *SVRSTUB. The default PROTOCOL(*TCP) is used to generate the server-side stub programs. The files are placed into the following file names:

- myrpc.h for a header file
- myrpc_xdr.c for an XDR file
- myrpc_clnt.c for client-side stubs
- myrpc_svc.c for server-side stubs

Example 2: Convert RPC Source to Client Stubs Only

```
CVTRPCSRC FROMFILE('/myrpc2') OPTION(*CLTSTUB)  
TOFILE('/myclnt.c')
```

This converts the RPC language file '/myrpc2' into client-side stub procedures. The results are placed into the file '/myclnt.c' as specified.

Error messages for CVTRPCSRC

*ESCAPE Messages

None.

RPCGEN (Convert RPC Source) Command

RPCGEN Command syntax diagram

RPCGEN Command	For the description of the RPCGEN command, see the CVTRPCSRC (Convert RPC Source) command description.
----------------	--

CVTTCPCL (Convert TCP/IP CL Source) Command Description

CVTTCPCL Command syntax diagram

Purpose

The Convert TCP/IP CL Source (CVTTCPCL) command is used to convert TCP/IP commands from releases prior to Version 3, Release 1, Modification 0 (V3R1M0) to the command syntax for the current release. The TCP/IP CL commands to be converted must exist in a source physical file.

The following commands are converted based on their specified parameter values. In some cases you may need to manually update the commands after conversion. Messages are issued to help identify the command statements that require manual updates.

ADDTCPLNK
CHGTCPLNK
RMVTCPLNK
STRTCPLNK
ENDTCPLNK
ADDTCPRTE
CHGTCPRTE
RMVTCPRTE
ADDTCPPORT
RMVTCPPORT
ADDTCPRSI
RMVTCPRSI
CHGTCPA
ENDTCPCNN
STRTCPTELN
CHGVT1MAP
SETVT1MAP
DSPVT1MAP
ENDSBS SBS(QTCP)
STRSBS SBS(QTCP/QTCP)

The CVTTCPCL command creates a printer file with the name CVTTCPCL. This printer file contains a report that indicates the success or failure of the source file conversion.

If a printer device file with the name CVTTCPCL is found in the job's library list when the CVTTCPCL command is issued, that printer device file is used to create the printer file. Otherwise, the CVTTCPCL command uses the Override with Printer File (OVRPRTF) command to use printer device file QSYS/QSYSPRT to create the printer file.

Note: Use the Create Printer File (CRTPRTF) command to create a printer device file.

Successful conversions of TCP/IP command source are noted in the report with the message:

TCP1E08 Member has been converted.

Informational messages are printed for unsuccessful command conversions. (Informational messages are also sent to the job log during conversion, and a single escape message is sent when the CVTTCPCL command has completed if any informational messages have been sent.) Some examples of functions that cannot be converted and may be printed as informational messages in the report are:

```
TCP1E07 Command &1 cannot be converted
TCP1E10 Parameter keyword cannot be converted
        in command &1
```

The user can write a program, perhaps by using the Copy Spooled File (CPYSPLF) command, to process the report based on the success or failure of the conversion.

Required Parameters

FROMFILE

Specifies the qualified name of the iSeries 400 CL source file containing TCP/IP commands to convert.

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.

from-file: Specify the name of the iSeries 400 CL source file to convert.

TOFILE

Specifies the qualified name of the file in which the converted source is placed. It must be different than the name of the FROMFILE parameter.

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.

to-file: Specify the name in which the converted source file is placed.

FROMMBR

Specifies the member name of the source file member to convert.

***ALL:** All members of the specified source file are converted to V3R1M0 TCP/IP command syntax if possible.

generic-member-name:* Specify the generic name of the source file members to convert. 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 converted only if *ALL or *ALLUSR library values can be specified for the name. Only the first 50 members matching the generic name's criteria will be converted.

member-names: Specify the names of the source file members to convert. Specify no more than 50 names.

Note that the only source members that are processed are those with a member type of CL, CLP, or TXT. Members in the FROMFILE with any other value for the member type are ignored by the CVTTCPCL command. If a source member is processed, the name of the converted source member in the TOFILE will be the same as the member name in the FROMFILE.

Example for CVTTCPCL

```
CVTTCPCL FROMFILE(OLDLIB/QCLSRC)
         TOFILE(NEWLIB/QCLSRC)
         FROMMBR(TCPPGM1 TCPPGM2 TCPPGM3)
```

This command converts all TCP/IP commands in the three members (TCPPGM1, TCPPGM2, TCPPGM3) of a CL source file (QCLSRC) located in library OLDLIB, to their new command names and formats. The converted source file members are located in QCLSRC, in library NEWLIB. The converted members keep their original member names, TCPPGM1, TCPPGM2, and TCPPGM3.

Error messages for CVTTCPCL

*ESCAPE Messages

CPF9801

Object &2 in library &3 not found.

CPF9810

Library &1 not found.

TCP1E02

File &1 in library &2 not found.

TCP1E03

File &1 in library &2 not a source file.

TCP1E06

Specified TOFILE same as FROMFILE.

COPY (Copy) Command Description

COPY command syntax diagram

COPY Command
For the description of the COPY command, see the CPY (Copy) command description.

CPY (Copy) Command Description

CPY Command syntax diagram

Purpose

The Copy (CPY) command copies a single object or a group of objects. By default, if the target object already exists, the copy of that individual object will fail. If the REPLACE(*YES) parameter is specified the

target object is overwritten. The newly created object must be renamed if it is stored in the same directory as the original object. If it is stored in a directory other than the one that contains the original object, it can retain the name of the original object.

An object name pattern can be used to copy a group of related objects. A pattern cannot be used to copy a group of objects from one file system to another unless the names in the source meet the requirements of the target file system. For example, a file named /OBJA in QOpenSys cannot be copied to directory /QSYS.LIB/MYLIB.LIB/FILEA.FILE, because the QSYS.LIB file system requires a name in the form OBJA.MBR when writing to a file. All names found within the pattern would fail if they did not meet the requirement of name.object-type.



The Copy command can also be used to copy a directory tree where the directory, its contents, and the contents of all of its subdirectories are copied. A subtree copy will attempt to preserve as many attributes from the original objects as possible. This would make it possible to migrate data from one file system to another.

If the original object is a read-only file, (a file that has the PC read-only attribute flag turned on), and SUBTREE(*NODIR) is specified, the newly created object will **not** be read-only. This follows the conventions of the OS/2 hierarchical file system (HFS).

Note:

When the value of the SUBTREE parameter is *NONE or *ALL, the PC read-only attribute flag will be copied. The subtree copy is intended to preserve as many attributes of the original objects as possible.

When TODIR is specified, the object is copied to that directory with the same name. The copied object is authorized the same as the original object. The user who issues the command owns the copied object if the OWNER value is *NEW.


When copying a file with SUBTREE(*NODIR) specified to the "root" (/). QOpenSys, QDLS, and UDFS file systems, the Last access date/time timestamp and the Data change date/time timestamp are preserved in the new file, and the Attribute change date/time timestamp is updated to the current time. The Last access date/time timestamp of the original file is updated to the current time. In the case of copying to a database file member (*MBR) in the QSYS.LIB  or independent ASP QSYS.LIB  file system, the Data change date/time is updated as well.

Note:

If the parameter SUBTREE(*YES) is specified the Create date/time is updated as well.

This command can also be issued using the following alternative command name:

- COPY

In addition to the CPY command, the Copy to Stream File (CPYTOSTMF) and Copy from Stream File (CPYFRMSTMF) commands can be used to copy between stream files and database member files or save files. 

File System differences:

- QSYS.LIB and independent ASP QSYS.LIB File System Differences:
 - If copying to a database file member from a different object type, or copying to or from a member not in the current job's library name space, some attributes are copied, see the Integrated file system topic in the File systems and management category of the Information Center for more information.
 - When copying a database member to another member within the same library name space, attributes are handled in the same manner as the Copy File (CPYF) command (this only applies if the Data Format parameter is *BINARY).

- Other object types are handled the way the Create Duplicate Object (CRTDUPOBJ) command handles attributes. (this only applies if the Data Format parameter is *BINARY)
- The REPLACE(*YES) option is only supported on file members, user spaces, and save files when the target object exists. All other object types will fail when the target object exists.
- QOPT File System Differences:
 - If copying a file within the QOPT file system, the Create date/time is always updated to the current time.
- QNetWare File System Differences:
 - If copying a file or directory to a location on the same server, the owner of the target object is always the caller of the command and the PC read-only flag is not copied.
- QFileSvr.400 File System Differences:
 - The OWNER(*KEEP) parameter is not supported when copying an object to the QFileSvr.400 File System. The copy will fail with error message CPFA0AD.
- Network File System (NFS) Differences:
 - The OWNER(*KEEP) parameter is not supported when copying an object to or from a mounted Network File System (NFS) directory. The copy will fail with error message CPFA0AD.



For more information about integrated file system commands, see the Integrated file system topic in the File systems and management category of the Information Center.

Restrictions:

1. The user must have object operational and read authorities and object management authority to the existing object.
2. The user must have execute authority to the directories in the path name prefixes.
3. The user must have add authority to the directory for the new object.
4. >> The user must have *WX authority in QOpenSys, *OBJMGT authority in QSYS.LIB and independent ASP QSYS.LIB, and *CHANGE authority in QDLS. QSYS.LIB and independent ASP QSYS.LIB also require the user to have *ADD authority to the parent of the parent of the new object.
5. The user must have authorization list management authority if the object is an authorization list.
6. The CPY command will copy the object's public and private authorities where it is supported.
7. If OWNER(*KEEP) is specified the user must have *ALLOBJ special authority or have add authority to the user profiles of the owners of the objects being copied.
8. If REPLACE(*YES) is specified the user must have *W, *OBJEXIST, and *OBJMGT to the target objects if they already exist.
9. >> The user must have *IOSYSCFG authority when copying character special files (*CHRSF objects).



Required Parameter

OBJ Specifies the path name of the object or a pattern to match the name of the object to be copied.

The object path name can be either a simple name or a name that is qualified with the name of the directory in which the object is located. A pattern can be specified in the last part of the path name. An asterisk (*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes. See path names for more information on specifying path names.

Note: An object name pattern can be used to copy multiple objects only when the TODIR parameter is specified.

Optional Parameters

TODIR

Specifies the path name of the directory to copy the object into. When this parameter is used, the copied object has the same name as the OBJ parameter specified.

..: The object is copied to the current directory with the same name as the existing object.

directory-path-name: Specify the path name of the existing directory to copy the object into. See path names for more information on specifying path names.

TOOBJ

Specifies the path name of the copied object. This is the name of the new object, including the path or the relative path. See path names for more information on specifying path names.

SYMLNK

Specifies whether to copy the object or a symbolic link to the object.

*NO: The object, not a symbolic link to the object, is copied.

Note:

If a symbolic link is encountered during the copy of a subtree, the object it points to is copied. If the symbolic link points to a directory, the directory is copied but its contents are not.

*YES: If the object being copied is a symbolic link, the symbolic link is copied, instead of copying the object that the symbolic link points to.

FROMCCSID

Specifies the method for obtaining the coded character set identifier (CCSID) for the source of the copy operation. This CCSID will be used for data conversion, if requested. This parameter is ignored if the object specified on the OBJ parameter is not a regular file. A regular file is a file that supports the integrated file system I/O operations open, read, and write.

*OBJ: Use the data CCSID of the object being copied.

*PCASCII: Use the data CCSID of the object being copied to compute a CCSID in the Microsoft Windows encoding scheme (x4105). Use this as the CCSID from which the data will be converted when DTAFMT(*TEXT) is specified. This option allows data from PCs to be converted properly, if the data was created using Microsoft Windows.

*JOBCCSID: The coded character set identifier (CCSID) from the default job CCSID is used.

from-CCSID: Specify a CCSID value between 1 and 65533.

TOCCSID

Specifies the data coded character set identifier (CCSID) for the target of the copy operation. This parameter is ignored if the object specified on the OBJ parameter is not a regular file. A regular file is a file that supports the integrated file system I/O operations open, read, and write.

*OBJ: Use the data CCSID of the object being copied. If this CCSID cannot be used by the file system that the object is being copied into, the copy operation will fail.

*CALC: Use the data CCSID of the object being copied. If this CCSID cannot be used by the file system that the object is being copied into, allow the file system to determine a different CCSID and continue with the copy.

***STDASCII:** Compute a CCSID in the IBM PC Data encoding scheme (x2100), based on the source file's CCSID. Associate this CCSID with the target of the copy operation and, if DTAFMT(*TEXT) is specified, also use this CCSID for the data conversion. If this CCSID cannot be used by the file system that the object is being copied into, the copy operation will fail.

***PCASCII:** Compute a CCSID in the Microsoft Windows encoding scheme (x4105), based on the source file's CCSID. Associate this CCSID with the target of the copy operation and, if DTAFMT(*TEXT) is specified, also use this CCSID for the data conversion. This option allows the resulting data to be used by Microsoft Windows applications. If this CCSID cannot be used by the file system that the object is being copied into, the copy operation will fail.

***JOBCCSID:** The coded character set identifier (CCSID) from the default job CCSID is used.

to-CCSID: Specify a CCSID value between 1 and 65533. If this code page cannot be used by the file system that the object is being copied into, the copy operation will fail.

FROMCODPAG

Specifies the method for obtaining the code page for the source of the copy operation. This code page will be used for data conversion, if requested. This parameter is ignored if the object specified on the OBJ parameter is not a regular file. A regular file is a file that supports the integrated file system I/O operations open, read, and write.

Note:

This keyword is replaced by FROMCCSID but the FROMCODPAG keyword can still be used. However, because this keyword may be removed in a future release, whenever possible use the FROMCCSID keyword.

***OBJ:** Use the data code page of the object being copied.

***PCASCII:** Use the data code page of the object being copied to compute a code page in the Microsoft Windows encoding scheme (x4105). Use this as the code page from which the data will be converted when DTAFMT(*TEXT) is specified. This option allows data from PCs to be converted properly, if the data was created using Microsoft Windows.

code_page: A code page value between 1-32767.

TOCODEPAGE

Specifies the data code page for the target of the copy operation. This parameter is ignored if the object specified on the OBJ parameter is not a regular file. A regular file is a file that supports the integrated file system I/O operations open, read, and write.

Note:

This keyword is replaced by TOCCSID but the TOCODEPAGE keyword can still be used. However, because this keyword may be removed in a future release, whenever possible use the TOCCSID keyword.

***OBJ:** Use the data code page of the object being copied. If this code page cannot be used by the file system that the object is being copied into, the copy operation will fail.

***CALC:** Use the data code page of the object being copied. If this code page cannot be used by the file system that the object is being copied into, allow the file system to determine a different code page and continue with the copy.

***STDASCII:** Compute a code page in the IBM PC Data encoding scheme (x2100), based on the source file's code page. Associate this code page with the target of the copy operation and, if DTAFMT(*TEXT) is specified, also use this code page for the data conversion. If this code page cannot be used by the file system that the object is being copied into, the copy operation will fail.

***PCASCII:** Compute a code page in the Microsoft Windows encoding scheme (x4105), based on the source file's code page. Associate this code page with the target of the copy operation and, if DTAFMT(*TEXT) is specified, also use this code page for the data conversion. This option allows the resulting data to be used by Microsoft Windows applications. If this code page cannot be used by the file system that the object is being copied into, the copy operation will fail.

code_page: A code page value between 1-32767. If this code page cannot be used by the file system that the object is being copied into, the copy operation will fail.

DTAFMT

Specifies the format of the data in the file being copied.

***BINARY:** The file contains data in binary format (such as an executable file).

Do not convert data on the copy. However, if the object being copied to has a different code page than the source object, all extended attributes will be converted into the code page of the new object before being set.

***TEXT:** The file contains data in textual form. Convert data to the code page of the new object during the copy. The data is processed as text during the copy.

If you are copying from a database member to a stream file, any line-formatting characters (such as carriage return, tab, and end-of-file) are just converted from one code page to another.

If you are copying from a stream file to a database member, the stream file must contain end-of-line characters or the copy will fail. If the stream file does contain end-of-line characters, the following actions are performed during the copy to a database file.

- End-of-line characters are removed.
- Records are padded with blanks (for a source physical file member) or nulls (for a data physical file member).
- Tab characters are replaced by the appropriate number of blanks to the next tab position.

SUBTREE

Specifies whether or not to copy a directory subtree if the object specified by OBJ is a directory.

***NODIR:** The object or objects specified by OBJ are copied. If an object is a directory the copy will fail.

***NONE:** The objects specified by OBJ are copied. Directory objects are copied but their contents are not copied.

***ALL:** The objects specified by OBJ are copied. Directory objects are copied as well as their contents and the contents of all subdirectories.

Pattern matching on the Object (OBJ) parameter only applies to the first level object. If the first level object is a directory, the pattern matching does not apply to its contents or the contents of its subdirectories.

If SUBTREE(*ALL) is specified, individual completion messages for each object are not issued. A final message is issued to indicate how many copies succeeded and how many failed. If objects did fail to copy the command will issue a diagnostic message for each copy that failed.

There are a few differences in how attributes are copied when SUBTREE(*NONE) or SUBTREE(*ALL) is specified instead of the default SUBTREE(*NODIR). A directory subtree copy preserves as much of the original objects attributes as possible.

- The PC read-only attribute flag is turned off in the copied object. If SUBTREE(*NONE) or SUBTREE(*ALL) is specified the flag will be copied.
- The Create date/time timestamp will be copied if SUBTREE(*NONE) or SUBTREE(*ALL) is specified (by default it is updated to the current time).

REPLACE

Specifies whether the target object is replaced if it already exists.

***NO:** The target object is not replaced if it already exists.



***YES:** If the target object already exists, it is replaced. If REPLACE(*YES) is specified with a directory object, the attributes of the existing target directory are changed but the objects that the directory contains are not removed.

OWNER

Specifies the owner of the newly created object.

***NEW:** The owner of the new object is the current user profile of the job.

***KEEP:** The owner of the new object is the same as the owner of the original object being copied.

Some file system objects do not support changing the owner. For example, the owner of *MBR objects in the QSYS.LIB  and independent ASP QSYS.LIB file systems will be  determined by the owner of the *FILE object that they are copied into.

Examples for CPY

Example 1: Copying a File

```
CPY OBJ('DECEMBER-1994-MONTHLY-PAYROLL-FILE')
    TOOBJ('PAY')
```

This command creates another file named PAY that is a duplicate of the file named DECEMBER-1994-MONTHLY-PAYROLL-FILE.

Example 2: Copying a File to Another Directory

```
CPY OBJ('PAY') TODIR('MYDIR')
```

This command creates another file named PAY in directory MYDIR.

Example 3: Copying a Symbolic Link

```
CPY OBJ('SL1') TOOBJ('YOURDIR/SL2') SYMLNK(*YES)
```

If SL1 is a symbolic link, the new object YOURDIR/SL2 is also a symbolic link. If SYMLNK(*NO) was specified, the new object would be a copy of whatever SL1 pointed to, as long as it was a legal candidate for the copy function.

Example 4: Copying with Conversion

```
CPY OBJ('/DATAFB')
    TOOBJ('/QSYS.LIB/APP1.LIB/DATA.FILE/DATAFB.MBR')
    TOCCSID(*CALC) DTAFMT(*TEXT)
```

The stream file 'DATAFB' is to be copied to the database file 'DATAFB.MBR'. By specifying TOCCSID(*CALC), the file system being copied to (the QSYS.LIB file system in this case) will try to create the new member in the same coded character set identifier (CCSID) as '/DATAFB'. If this fails (in this case, if 'DATA.FILE' is not in the same CCSID as 'DATAFB'), the file system will be allowed to choose an

appropriate CCSID and complete the copy. By specifying DTAFMT(*TEXT), the data in 'DATAFB' is handled as text and is converted into the CCSID chosen for the new file 'DATAFB.MBR'.

Example 5: Copying a Directory Subtree

```
CPY   OBJ('/QDLS/MYINFO')
      TODIR('/myfolder')
      SUBTREE(*ALL)
      OWNER(*KEEP)
      REPLACE(*YES)
```

The *FLR object (QDLS file system folder) is created in the '/myfolder' directory in the "root" (/) file system with the path name '/myfolder/MYINFO'. Its contents are copied as well. Since OWNER(*KEEP) is specified, the new objects created will belong to the same profiles as the old objects. With the REPLACE parameter set to *YES if any of the target files already exist they will be overwritten.

Error messages for CPY

*ESCAPE Messages

CPFA082

*ADD authority required to owner's user profile.

CPFA083

Insufficient authority to replace object. Object is &1.

CPFA085

Home directory not found for user &1.

CPFA08E

More than one name matches pattern.

CPFA093

Name matching pattern not found.

CPFA09C

Not authorized to object.

CPFA09D

Error occurred in program &1.

CPFA0A1

An input or output error occurred.

CPFA0A3

Path name resolution causes looping.

CPFA0A6

Number of links exceeds maximum allowed for the file system.

CPFA0A7

Path name too long.

CPFA0A9

Object not found.

CPFA0AA

Error occurred while attempting to obtain space.

CPFA0AB

Object name not a directory.

CPFA0AD

Function not supported by file system.

CPFA0B0

Request not allowed to operate from one file system to another.

CPFA0B2

No objects satisfy request.

CPFA0BB

&1 objects copied. &2 objects failed.

CPFA0C4

Object name not a file.

CPFA0DA

Object name is a directory.

CPFB41E

Object type must match replaced object type.

CPYCFGL (Copy Configuration List) Command Description

CPYCFGL Command syntax diagram

Purpose

The Copy Configuration List (CPYCFGL) command creates a configuration list as a copy of an existing configuration list.

Required Parameters

FROMCFGL

Specifies the configuration list from which to copy. APPN local and remote location lists and asynchronous location lists are not allowed for copying. Only one of these list types is allowed per system.

CFGL Specifies the name of the configuration list being created.

Optional Parameters

AUT Specifies the authority given to users who do not have specific authority to the configuration list, who are not on an authorization list, and whose user group has no specific authority to the configuration list.

***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 change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***USE:** The user can perform basic operations on the configuration list, such as running a program or reading a file. The user cannot change the configuration list. *USE authority provides object operational authority, read authority, and execute authority. If the object is an authorization list, the user cannot add, change, or remove users.

***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 also can change ownership of the configuration list.

***EXCLUDE:** The user cannot access the configuration list.

TEXT Specifies the text that briefly describes the configuration list. More information on this parameter is in Commonly used parameters.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example for CPYCFGL

```
CPYCFGL FROMCFGL(CONFIG01) CFGL(CONFIG02)
```

This command copies the configuration list named CONFIG01 to a new configuration list name CONFIG02.

Error messages for CPYCFGL

***ESCAPE Messages**

CPF2182

Not authorized to library &1.

CPF260D

Configuration list &1 already exists.

CPF260E

Configuration list &1 not created.

CPF260F

Configuration list &1 not found.

CPF2612

List type &1 not correct for copy.

CPF2625

Not able to allocate object &1.

CPF2634

Not authorized to object &1.

CPF2663

Configuration list &1 previously deleted.

CPYIGCTBL (Copy DBCS Font Table) Command Description

CPYIGCTBL Command syntax diagram

Purpose

The Copy DBCS Font Table (CPYIGCTBL) command copies part or all of a double-byte character set (DBCS) font table to tape, diskette, or physical file; or from tape, diskette, or physical file into the font table. Copying a DBCS font table from tape, diskette, or physical file into a font table also puts its definition in the system. DBCS font tables are objects and can be saved and restored.

DBCS font tables contain the images in a given dot matrix of the DBCS extension characters used on the system. The system refers to the tables when printing and displaying these characters. There are separate tables for each character image matrix used by devices attached to the system.

Restriction: A physical file used to save and restore table information must have a minimum record length of 74 bytes.

Required Parameters

IGCTBL

Specifies the name of the DBCS font table being copied. Choose one of the following table names:

QIGC2424: The Japanese DBCS font table used for displaying and printing extension characters in a 24-by-24 dot matrix image.

QIGC2424C: The Traditional Chinese DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC2424K: The Korean DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC2424S: The Simplified Chinese DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC3232: The Japanese DBCS font table used for displaying and printing extension characters in a 32-by-32 dot matrix image.

QIGC3232S: The Simplified Chinese DBCS font table used for printing extension characters in a 32-by-32 dot matrix image.

QIGCrrcccl: The name of the DBCS font table to be copied must always be in the format QIGCrrcccl, where *rr* is the table row matrix size, *cc* is the table column matrix size, and the letter *l* is an optional language identifier.

OPTION

Specifies how to copy the DBCS font tables: either from the system to diskette, tape, or physical file; or from diskette, tape, or physical file into the system.

***OUT:** The specified DBCS font table is copied to diskette, tape, or physical file.

***IN:** The specified DBCS font table is copied from diskette, tape, or physical file to the system.

DEV Specifies whether a device (tape or diskette) or a physical file is used to save or restore tables.

***FILE:** Specifies that the DBCS font table is saved to, or restored from, a physical file.

device-name: Specifies the name of the tape or diskette device that the table is saved to, or restored from. The device name must already be known on the system by a device description.

LABEL

Specifies the object that identifies the data file on diskette or tape that contains the DBCS font table. When copying the table into the system, the label identifies the file that exists on diskette or tape. When copying the table to diskette or tape, the label identifies the file that is created on diskette or tape.

***IGCTBL:** Identifies the data file on diskette or tape that contains the DBCS font table. It is the same as the name specified on the IGCTBL parameter without the first character.

data-file-identifier: Specify the identifier (8 characters maximum for diskette and 17 maximum for tape, starting with an alphabetic character) of the data file that is used with this DBCS font table.

Optional Parameters

SELECT

Specifies which portion of the DBCS font table is to be copied.

***ALL:** All IBM-supplied and user-defined DBCS characters are copied.

***SYS:** Only IBM-supplied DBCS characters are copied.

***USER:** Only user-defined DBCS characters are copied.

***RANGE:** Only user-defined DBCS characters that fall in the range specified on the RANGE parameter are copied.

RANGE

Specifies the range of user-defined characters to be copied. Specify this value only when choosing SELECT(*RANGE).

The DBCS codes and numbers that can be specified for range values are listed in tables at the end of this parameter description. Tables are provided for the Japanese, Korean, traditional Chinese, and simplified Chinese languages.

Element 1: Starting Character to be Copied

The first of two values specified with the RANGE parameter specifies the first character to be copied. Specify one of the following values:

***FIRST:** The system starts copying with the first user-defined DBCS character in the table.

from-user-character: The system starts copying with the specified DBCS code or number. Specify the 4-character DBCS code, or the 5-character DBCS number.

Element 2: Ending Character to be Copied

The last of two values specified with the RANGE parameter specifies the last character to be copied. Specify one of the following values:

***LAST:** The system stops copying with the last user-defined character found.

to-user-character: The system stops copying with the specified DBCS code or number. Specify the 4-character DBCS code, or the 5-character DBCS number.

Following are tables that list the valid codes and numbers to specify for starting and ending values of user-defined character ranges.

Table 1. Japanese DBCS Codes for User-Defined Characters (RANGE parameter)

6941 - 69FE	6A41 - 6AFE	6B41 - 6BFE
6C41 - 6CFE	6D41 - 6DFE	6E41 - 6EFE
6F41 - 6FFE	7041 - 70FE	7141 - 71FE
7241 - 72FE	7341 - 73FE	7441 - 74FE
7541 - 75FE	7641 - 76FE	7741 - 77FE
7841 - 78FE	7941 - 79FE	7A41 - 7AFE
7B41 - 7BFE	7C41 - 7CFE	7D41 - 7DFE
7E41 - 7EFE	7F41 - 7FFE	

Table 2. Japanese DBCS Numbers for User-Defined Characters (RANGE Parameter)

10561 through 10750	10817 through 11006
11073 through 11262	11329 through 11518
11585 through 11774	11841 through 12030
12097 through 12286	12353 through 12542
12609 through 12798	12865 through 13054
13121 through 13310	13377 through 13566
13633 through 13822	13889 through 14078
14145 through 14334	14401 through 14590
14657 through 14846	14913 through 15102
15169 through 15358	15425 through 15614
15681 through 15870	15937 through 16126
16193 through 16382	

Table 3. Korean DBCS Codes for User-Defined Characters (RANGE Parameter)

D441 - D4FE	D541 - D5FE	D641 - D6FE
D741 - D7FE	D841 - D8FE	D941 - D9FE
DA41 - DAFE	DB41 - DBFE	DC41 - DCFE
DD41 - DDFE		

Table 4. Korean DBCS Numbers for User-Defined Characters (RANGE Parameter)

37953 through 38142	38209 through 38393
38465 through 38654	38721 through 38910
38977 through 39166	39233 through 39422
39489 through 39678	39745 through 39934
40001 through 40190	40257 through 40446

Table 5. Traditional Chinese DBCS Codes for User-Defined Characters (RANGE Parameter)

D041 - D0FE	D141 - D1FE	D241 - D2FE
D341 - D3FE	D441 - D4FE	D541 - D5FE
D641 - D6FE	D741 - D7FE	D841 - D8FE
D941 - D9FE	DA41 - DAFE	DB41 - DBFE
DC41 - DCFE	DD41 - DDFE	

Table 6. Traditional Chinese DBCS Numbers for User-Defined Characters (RANGE Parameter)

36929 through 37118	37185 through 37374
37441 through 37630	37697 through 37886
37953 through 38142	38209 through 38398
38465 through 38654	38721 through 38910
38977 through 39166	39233 through 39422
39489 through 39678	39745 through 39934
40001 through 40190	40257 through 40446

Table 7. Simplified Chinese DBCS Codes for User-Defined Characters (RANGE Parameter)

7641 - 76FE	7741 - 77FE
7841 - 78FE	7941 - 79FE
7A41 - 7AFE	7B41 - 7BFE
7C41 - 7CFE	7D41 - 7DFE
7E41 - 7EFE	7F41 - 7FFE

Table 8. Simplified Chinese DBCS Numbers for User-Defined Characters (RANGE Parameter)

13889 through 14078	14145 through 14334
14401 through 14590	14657 through 14846
14913 through 15102	15169 through 15358
15425 through 15614	15681 through 15870
15937 through 16126	16193 through 16382

RPLIMG

Specifies whether user-defined DBCS characters in the specified table are replaced with those found on tape or diskette. Specify this value only when choosing OPTION(*IN).

***NO:** The system does not replace user-defined DBCS characters in the table stored in the system with those found on tape or diskette.

***YES:** The system replaces user-defined DBCS characters in the table stored in the system with those found on tape or diskette.

VOL Specifies one or more volume identifiers used by the file. More information on this parameter is in Commonly used parameters.

***MOUNTED:** The volume currently placed in the device is used.

volume-identifier: Specify the identifiers of one or more diskettes or tapes in the order in which they are placed in a tape or diskette drive and used in the copy operation.

EXPDATE

Specifies, when OPTION(*OUT) is indicated, the end date of the file on diskette or tape that contains the DBCS font table. If a date is specified, the file is protected and cannot be written over until the day after the specified end date.

***PERM:** The data file is permanently protected. An expiration date of 999999 is assigned.

expiration-date: Specify the expiration date after which the data file is no longer protected. The date must be specified in the format defined by the system values QDATFMT and, if separators are used, QDATSEP. However, the specified date is put in the diskette or tape label as yymmdd, where yy is the year, mm is the month, and dd is the day.

SEQNBR

Specifies (only when tape is used) which sequence number is used as the starting point for the copy process. This value is valid only if the parameter OPTION(*OUT) is specified.

Note:

This value is valid only if *OUT is specified on the OPTION parameter.

***END:** The system saves the table after the last sequence number on the tape.

***SEARCH:** The volume that is in the tape or diskette unit is searched for a data file with an identifier that compares with the LABEL parameter value; when a match is found, the table is restored. If the last operation on the device specifies ENDOPT(*LEAVE) (the tape is positioned at the location at which the last operation ended), the file search starts with the first data file beyond the current tape position. If ENDOPT(*LEAVE) was not used for the last operation (or if the tape was manually rewound since an ENDOPT(*LEAVE) operation), the search starts with the first data file on the volume. This value is valid only if the parameter OPTION(*IN) is specified.

file-sequence-number: Specify the sequence number of the file that is used. Valid values range from 1 through 9999.

ENDOPT

Specifies the operation that is automatically performed on the tape volume after the operation ends. If more than one volume is included, this parameter applies only to the last tape volume used; all other tape volumes are rewound and unloaded when the end of the tape is reached.

***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.

FILE Specifies the name of the existing physical file that contains the DBCS font table.

The name of the physical file can be qualified by the following library value:

library-name: Specify the name of the library to be used.

physical-file-name: Specify the name of the physical file.

MBR Specifies the name of the existing physical file member that the table is saved to, or restored from.

Example for CPYIGCTBL

```
CPYIGCTBL IGCTBL(QIGC2424) OPTION(*OUT) LABEL(*IGCTBL)
DEV(QDKT)
```

This command causes the system to copy the complete Japanese DBCS font table QIGC2424 from the system to the diskette. The name of the label on the diskette is IGC2424.

Additional Considerations

Consider the following before entering this command:

- The diskette used in the copy operation must be in the *DATA format. The Basic System Operations topic in the Information Center has instructions on initializing diskettes in the *DATA format.
- The system creates the DBCS font table in addition to copying it when you specify OPTION(*IN), if the following is true:
 - The specified table does not already exist in the system.
 - The tape or diskette that you are copying the table from contains all of the DBCS characters supplied with your system.
 - SELECT(*ALL) or SELECT(*SYS) was specified.
- Consider copying a DBCS font table to tape or diskette before deleting that table from the system.

Error messages for CPYIGCTBL

*ESCAPE Messages

CPF8181

DBCS font table &4 damaged.

CPF8416

DBCS font table &1 not updated and no images copied.

CPF8417

Error found in RANGE keyword.

CPF8418

Data file &2 cannot be used to copy DBCS font table &1.

CPF8419

DBCS font table &1 not created and no images copied.

CPF8420

CPYIGCTBL command ended due to error.

CPF8421

DBCS font table &1 not found.

CPF8422

Not able to use DBCS font table &1.

CPF8423

Error found in keyword IGCTBL.

CPF8426

Device &1 either not found, or not valid for command.

CPF8427

DBCS font table &1 not migrated.



CPYDSTRPSO (Copy Distribution Repository Object) Command

Description

Note: To use this command, you must have the 5722-MG1 (Managed System Services for iSeries) licensed program installed.

CPYDSTRPSO Command syntax diagram

Purpose

The Copy Distribution Repository Object (CPYDSTRPSO) command is used to copy an object residing in the distribution repository to a library, folder, or stream file. The object type of the object specified must match the information in the distribution repository.

Required Parameter

GLBNAME

Specifies the tokens of the global name used to find the unique catalog entry to be copied. A global name should consist of at least two tokens.

Element 1: Token 1

***NETID:** The first global name token value is a network ID generated by the command from the network attributes.

global-name-token-1: Specify the first token of the global name.

The first token must be the registered enterprise ID or network ID. The values of tokens 2 through 10 are assigned by the authority identified by the name in the first token.

Element 2-10: Token 2-10

***ANY:** The corresponding token value is ignored.

***HIGHEST:** The catalog entry with the highest corresponding token value is selected. The version attribute of the token must be: *ORDCHAR, *ORDDATE, *ORDDEC, or *ORDTIME.

***LOWEST:** The catalog entry with the lowest corresponding token value is selected. The version attribute of the token must be: *ORDCHAR, *ORDDATE, *ORDDEC, or *ORDTIME.

***CPNAME:** The global name token value is a control point name. This value is generated from the network attributes.

***NETID:** The global name token *n* value is a network ID. This value is generated from the network attributes.

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

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

global-name-token-n: Specify a token of the global name. This field is left adjusted and padded on the right with blanks.

Optional Parameters

OBJ Specifies the object and library into which the repository object is copied. The distribution repository object cannot be copied into the QTEMP library.

The name of the repository object 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.

object-name: Specify the name of the object into which the repository object is copied.

MBR Specifies the file member name into which the repository object is copied. This value is valid when the distribution repository object type is a file.

***ALL:** The entire file is copied into the file specified on the OBJ parameter. This value is not valid for object type *FILEDATA.

***FIRST:** The first member in the file receives the member. If *FIRST is specified and a member does not exist, a member is created with the name of the file specified on the OBJ parameter.

member-name: Specify the name of the file member into which the data is copied.

DLO Specifies the document name into which the distribution repository object is copied. The document name and the folder path must be specified to copy a document. Only folder name (FLR parameter) is required to copy a folder. This parameter is valid only when the distribution repository object being copied is a document.

FLR Specifies the folder name into which the distribution repository object is copied. This parameter is required when the distribution repository object being copied is a document or a folder.

folder-name: Specifies the name of the folder path in which the documents are copied (for object type *DOC) or folder name is to be copied (for object type *FLR).

STMF Specifies the stream file into which the distribution repository object is copied.

object-path-name: Specifies the path name of the stream file into which the data is copied. The length of the path name can be up to 5000 characters. This parameter can be specified only when the distribution repository object being copied contains a stream file (*STMF), a file data (*FILEDATA), or a non-supported OS/400 object type. Additional information about path names is in the Integrated File System Introduction topic in the Information Center.

REPLACE

Specifies whether existing data is replaced or add records to the file members.

***NO:** Existing objects remain unchanged. New objects are created with the data.

***YES:** Existing objects are replaced with the new data. New objects are created and the user is given the proper authority to the object.

***ADD:** Existing members are added to the new records after the existing members. When a target file member does not exist, it will be created. The file that contains the members must exist.

Examples for CPYDSTRPSO

Example 1: Copying a Distribution Repository File

```
CPYDSTRPSO GLBNAME(ABC *LOWEST *HIGHEST *ANY ROLLM)
OBJ(*LIBL/MYFILE) REPLACE(*YES)
```

This command copies a file from the distribution repository to the local object MYFILE. The file must meet the search criteria for the global name specified. If the object exists, it is replaced. If more than one catalog entry matches, the request is rejected.

Example 2: Copying a Distribution Repository Document

```
CPYDSTRPSO GLBNAME(NETID *LOWEST *HIGHEST *ANY ROLLM)
           DLO(STATUS) FLR(MNPS1/SALES.APR) REPLACE(*NO)
```

This command copies a document from the distribution repository to the document STATUS in folder path MNPS1/SALES.APR. The document must meet the search criteria for the global name specified. If the document exists, the copy is rejected. If the document is new, it is created. The folder path must already exist. If more than one catalog entry matches the search criteria, the request is rejected.

Example 3: Copying a Distribution Repository Folder

```
CPYDSTRPSO GLBNAME(NETID *LOWEST *HIGHEST *ANY ROLLM)
           FLR(MNPS1/SALES.APR) REPLACE(*YES)
```

This command copies a folder from the distribution repository to the folder MNPS1/SALES.APR. The folder must meet the search criteria for the global name specified. If the folder exists, it is replaced with the new information. If the folder does not exist it is created. If more than one catalog entry matches the search criteria, the request is rejected.

Example 4: Copying a Stream File to the Root File System

```
CPYDSTRPSO GLBNAME(STREAM FILE EXAMPLE)
           STMF('/Dir1/Dir2/Dir3/UsrFile')
           REPLACE(*YES)
```

This command copies an object residing in the distribution repository to a stream file specified in the object path name. If the stream file exists, it is replaced with the new information. If the stream file does not exist, it is created. If more than one catalog entry matches the search criteria, the request is rejected.

Example 5: Adding Members When Copying a File

```
CPYDSTRPSO GLBNAME(ADD FILE MEMBERS)
           OBJ(MYLIB/MYFILE) MBR(MYMBR)
           REPLACE(*ADD)
```

This command adds records to the MYLIB/MYFILE/MYMBR member.

Error messages for CPYDSTRPSO

*ESCAPE Messages

CPF2111

Library &1 already exists.

CPF2112

Object &1 in &2 type *&3 already exists.

CPF2132

Object &1 already exists in library &2.

CPF2146

Owner of object &1 and object being replaced not the same.

CPF2176

Library &1 damaged.

CPF2232

Not authorized to user profile &1.

- CPF2233**
No delete authority to user profile &1.
- CPF2451**
Message queue &1 is allocated to another job.
- CPF2802**
From-file &1 in &2 not found.
- CPF2813**
File &1 in &2 not available.
- CPF2861**
To-file &1 in &2 not found or not created.
- CPF2868**
Member or label not found or suspended in from-file.
- CPF3130**
Member &2 already in use.
- CPF3204**
Cannot find object needed for file &1 in &2.
- CPF3218**
Cannot allocate object needed for file &1 in &2.
- CPF3731**
Cannot use &2 &1 in library &3.
- CPF3733**
&2 &1 in &3 previously damaged.
- CPF3737**
Save and restore data area &1 not found.
- CPF3738**
Device &1 used for save or restore is damaged.
- CPF3761**
Cannot use &2 &1 in &3.
- CPF3764**
&2 &1 in &3 not found.
- CPF3767**
Device &1 not found.
- CPF3780**
Specified file for library &1 not found.
- CPF3781**
Library &1 not found.
- CPF3812**
Save file &1 in &2 in use.
- CPF4128**
Not able to allocate objects needed for file &2 in library &3 member or program device &4.
- CPF5729**
Not able to allocate object &1.
- CPF5813**
File &1 in library &2 already exists.

CPF7302
File &1 not created in library &2.

CPF81xx
Damaged object error messages.

CPF8A14
&2 of type &4 not renamed to &3 in folder &1.

CPF8A23
Folder &1 is not empty.

CPF8A75
Not authorized to access folder &1.

CPF8A77
Folder &1 not found.

CPF8A78
Folder &1 in use.

CPF8A80
Document &2 in use in folder &1.

CPF8A82
Document &2 not found in folder &1.

CPF8A83
Not authorized to access document &2 in folder &1.

CPF8A87
Document name &2 not correct.

CPF8A88
Operation not allowed on document &2 in folder &1.

CPF8A89
Document &2 in folder &1 is logically damaged.

CPF8A97
Folder name &1 not correct.

CPF9005
System resource required to complete this request not available.

CPF9006
User not enrolled in system distribution directory.

CPF9009
System requires file &1 in &2 be journaled.

CPF9012
Start of document interchange session not successful for &1.

CPF901F
*AUTL was specified for a user other than *PUBLIC.

CPF9029
Not allowed to specify owner profile &1.

CPF9031
No authority to specify DLO(*ALL).

CPF9032
Document interchange session not started.

- CPF903A**
Document or folder activity not stopped, requested operation cannot be done.
- CPF9048**
Ownership of &2 document library objects changed to profile &1; &3 not changed.
- CPF9073**
No authority to view or change the security of document library object &1.
- CPF908A**
Requester &1 not enrolled.
- CPF908E**
&1 objects changed; &2 objects not changed.
- CPF909A**
Document &2 in folder &1 is damaged.
- CPF9095**
Folder &1 is damaged.
- CPF90B8**
No authority to specify a reference object for document library object &1.
- CPF9801**
Object &2 in library &3 not found.
- CPF9803**
Cannot allocate object &2 in library &3.
- CPF9804**
Object &2 in library &3 damaged.
- CPF9807**
One or more libraries in library list deleted.
- CPF9808**
Cannot allocate one or more libraries on library list.
- CPF9809**
Library &1 cannot be accessed.
- CPF9810**
Library &1 not found.
- CPF9811**
Program &1 in library &2 not found.
- CPF9812**
File &1 in library &2 not found.
- CPF9814**
Device &1 not found.
- CPF9830**
Cannot assign library &1.
- CPF9831**
Cannot assign device &1.
- CPF9838**
User profile storage limit exceeded.
- CPF9845**
Error occurred while opening file &1.

CPF9846
Error while processing file &1 in library &2.

MSS0111
Distribution repository object not available.

MSS0114
Not authorized to distribution catalog.

MSS0116
Maximum global name length exceeded.

MSS0117
Global name token &3 not valid. Reason code &4.

MSS011B
Distribution catalog entry not found.

MSS011C
Distribution catalog not available.

MSS011D
More than one distribution catalog entry found.

MSS0123
Internal processing error occurred.

MSS0124
Error while managing distribution catalog.

MSS0125
Distribution catalog damaged.

MSS0151
Distribution repository object not copied.

MSS0153
Distribution repository object does not exist.

MSS0154
Object &1 already exists.

MSS0157
Not authorized to copy repository object.

MSS0158
Document &1 already exists.

MSS0159
Member &1 already exists.

MSS015D
REPLACE(*YES) not valid with specified global name.

MSS015E
Library name required.

MSS015F
Data not copied.

MSS0161
Library QTEMP not valid.

MSS0162
Object types do not match.

- MSS0163**
Distribution repository object not found.
- MSS0164**
Record length for file &2 too short.
- MSS0165**
Object cannot be decompressed.
- MSS0166**
Folder already exists.
- MSS0168**
Repository object contains more than one member.
- MSS016D**
Object could not be decompressed.
- MSS016E**
MBR(*ALL) not valid with object type &1.
- MSS0175**
REPLACE(*ADD) only valid for physical files.
- MSS019A**
Repository object not copied.
- MSS01D1**
Library must be QSYS.
- MSS01D3**
Object types do not match.
- MSS01D6**
Length of global name token &3 not valid.
- MSS01D7**
Value of global name token &3 not valid.
- MSS01D8**
Global name not valid.

CPYDOC (Copy Document) Command Description

CPYDOC Command syntax diagram

Purpose

The Copy Document (CPYDOC) command allows the user to copy a document from a folder into another folder, to copy a document that is not in a folder into a folder, and copy a document into the same folder or another folder with a different name.

The document copy is not indexed, regardless of whether or not the original document is indexed. If the document copy already exists and is already indexed, the index entry will not match the new content of the document copy, as the document is not reindexed. If you want the document copy to be indexed or reindexed, use the Add Text Index Entry (ADDTXTIDX) command after doing the copy.

Restrictions:

1. If the user is replacing a document, then the user must have *CHANGE authority for that document.
2. If the user is creating a new document, then the user must have *CHANGE authority for the folder that is to contain it. The new document is to have the same authorization as the document from which it is copied.

3. The user must have use authority for the from-document.

Required Parameter

FROMDOC

Specifies the name of the document being copied.

Note:

If FROMDOC(*document-name*) is specified, a folder name must be specified on FROMFLR. If FROMDOC(*SYSOBJNAM) is specified, a system object name must be specified on SYSOBJNAM.

***SYSOBJNAM:** A system object name is used to identify the document being copied.

document-name: Specify the name of the document that is copied.

Optional Parameters

FROMFLR

Specifies the name of the folder that contains the document to copy.

***NONE:** A folder name is not specified for the document. FROMFLR(*NONE) must be specified if the document is not in a folder. FROMFLR(*NONE) cannot be specified if a document name is specified for FROMDOC.

folder-name: Specify the name of the folder from which the document is copied.

TODOC

Specifies the folder into which the output document is copied.

Note:

The user cannot specify both TODOC(*FROMDOC) and TOFLR(*FROMFLR) to designate the copied document in its respective folder.

***FROMDOC:** The name of the new document is the same as the name specified on the FROMDOC parameter.

document-name: Specify the name of the new output document.

TOFLR

Specifies the folder into which the document is copied.

***FROMFLR:** The folder name is the same as that specified in the FROMFLR parameter and the document is copied into the same folder.

folder-name: Specify the name of the folder into which the document is copied.

REPLACE

Specifies whether the document specified by TODOC can be replaced. More information on this parameter is in Commonly used parameters.

***NO:** A new document, specified on the TODOC parameter, is created as a copy of the document being copied. If a document already exists in the folder specified on the TOFLR parameter, no copy is made.

***YES:** The document replaces an existing document with the same name in the folder specified on the TOFLR parameter. If no document with the same name exists in the folder, a new document is created.

SYSOBJNAM

Specifies the system object name. This parameter is valid only when DLO(*SYSOBJNAM) or DOCL(*SYSOBJNAM) is specified. A full ten characters must be specified.

Examples for CPYDOC

Example 1: Copying a Document

```
CPYDOC FROMDOC(MYDOC) FROMFLR(MYFLR)
      TODOC(MYDOC2) TOFLR(MYFLR2) REPLACE(*YES)
```

This command copies document MYDOC located in folder MYFLR to document MYDOC2 located in folder MYFLR2. If document MYDOC2 already exists in MYFLR2, the system replaces it with a copy of document MYDOC; otherwise, MYDOC2 is created in MYFLR2 as a copy of MYDOC in MYFLR.

Example 2: Copying Document and Keeping Source Document Name

```
CPYDOC FROMDOC(*SYSOBJNAM) SYSOBJNAM(AMBT133080)
      TODOC(MYDOC4) TOFLR(MYFLR)
```

This command copies a document, identified by the system object name, to document MYDOC4 located in folder MYFLR. The document name will be the same as the name of the source document.

Example 3: Copying Document to Document in Same Folder

```
CPYDOC FROMDOC(XYZ) FROMFLR('MYFLR/TEST')
      TODOC(NEW)
```

This command copies document XYZ located in folder MYFLR/TEST to document NEW in the same folder. If document NEW already exists, an error message is sent.

Error messages for CPYDOC

***ESCAPE Messages**

CPF8A12

Document &2 in folder &1 not copied.

CPYF (Copy File) Command Description

CPYF Command syntax diagram

Purpose

The Copy File (CPYF) command copies all or part of a database or external device file to a database or external device file. Copy operations can be done from physical, logical, diskette, tape, inline, or DDM files to physical, diskette, tape, printer, or DDM files.

Note:

For more information on the use of DDM files, see the Distributed Data Management topic in the Information Center.

Some of the specific functions that can be performed by the CPYF command include the following:

- Copying one database file member, a set of members with a common generic name, or all members in a database file. Many database file members can be copied to any valid to-file type except to more than one label on tape (FROMMBR and TOMBR parameters).

- Copying one diskette file label identifier, a set of labels with a common generic name, or all labels from a diskette file. Many diskette labels can be copied to any valid to-file type except to more than one label on tape (FROMMBR and TOMBR parameters).
- Adding records to an existing physical file member or replace the contents of a receiving physical file member (MBROPT parameter).
- Selecting certain records to be copied using one of the following methods:
 - Selecting a particular format in a logical file that has more than one record format (RCDFMT parameter).
 - Limiting the range of records copied based on starting and ending relative record numbers (FROMRCD and TORCD parameters).
 - Limiting the range of records copied based on starting and ending record key values (FROMKEY and TOKEY parameters).
 - Limiting the maximum number of records copied based on a number of records (NBRRCDS parameter).
 - Comparing a specified character string to a position in a record or field, or scan the record or field for the character string (INCCHAR parameter).
 - Comparing a specified value to one or more fields in each record. The record is selected based on a logical expression over the field comparisons (INCREL parameter).
- Copying records between database files whose from-file and to-file record formats are different, and converting records when the from-file and to-file are different file types (source and data, FMTOPT parameter).
- Inserting new sequence numbers and a zero date in the sequence number and date fields of records copied to a source physical file (SRCOPT parameter).
- Selecting a printout of either character or hexadecimal format (OUTFMT parameter).

Several other commands offer copy functions with a more specific set of parameters that are tailored to the to-file or from-file. For more information, refer to the following command descriptions:

CPYFRMDKT (Copy from Diskette)

CPYFRMTAP (Copy from Tape)

CPYSRCF (Copy Source File)


CPYTODKT (Copy to Diskette)

CPYTOTAP (Copy to Tape)

CPYFRMQRYF (Copy from Query File)

Table 1 (188), Table 2 (189), and Table 3 (189), at the end of this command description, show the parameters that are valid when copying from or to database files and device files.

Error Handling: The escape message CPF2817 is sent for many different error conditions that can occur during a copy operation. At least one diagnostic message that indicates the specific error condition always comes before the escape message. More information on handling errors is in the File Management topic in the Information Center.

Status Message: During the running of the CPYF command, message CPI2801 is sent as a status message informing the interactive user that a copy is occurring. More information on preventing status messages from appearing is in the CL Programming  book.

Restrictions:

1. During the time a CPYF request is run, the file named on the TOFILE parameter may be locked (similar to an *EXCL lock with no timeout) so that no access is possible.
2. When the CRTFILE(*YES) parameter is specified and the file copied (FROMFILE parameter) has an associated trigger, the file created (TOFILE parameter) does not have the associated trigger. The Add Physical File Trigger (ADDPFTRG) command must be used to add a trigger to the file.
3. In multithreaded jobs, this command is not threadsafe when copying from or to multiple database file members, device files (except SPOOL(*YES) print files), distributed files, or DDM files of type *SNA. This command fails for distributed files that use relational databases of type *SNA and DDM files of type *SNA. It is threadsafe ONLY when copying from and to single database file members (local or DDM of type *IP) or SPOOL(*YES) print files.

Required Parameters

FROMFILE

Specifies the qualified name of the database, inline data file, or device file that contains the records being copied. A database file can be a physical or logical file. A device file can be a diskette file or tape file.

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.

file-name: Specify the name of the file that contains the records to be copied.

TOFILE

Specifies the qualified name of the file that receives the copied records.

Note:

A device file can be a diskette file, tape file, or printer file. However: (1) If the from-file and to-file are both diskette files, the to-file must be spooled (SPOOL(*YES) must be specified on the Create Diskette File (CRTDKTF), Change Diskette File (CHGDKTF), or Override Diskette File (OVRDKTF) command). (2) An externally described printer file cannot be specified.

If the device file is a printer file or if TOFILE(*PRINT) is specified, shift-out shift-in (SO-SI) character pairs are not added around the graphic data. UTFMT(*HEX) can be specified to print the data in hexadecimal format.

***PRINT:** The data is copied to the IBM-supplied printer device file QSYSPRT and the file is formatted according to the UTFMT parameter.

The IBM-supplied printer file QSYSPRT may not be overridden to a different file name, and it must have the RPLUNPRT(*YES) and CTLCHAR(*NONE) attributes.

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.

file-name: Specify the qualified name of the physical file or device file that receives the copied records. If no library qualifier is specified, *LIBL is used to locate the file. However, if CRTFILE(*YES) is specified and the specified file cannot be found, the file name must be qualified with a library name. When the physical to-file is created, it is placed in the specified library.

Optional Parameters

FROMMBR

Specifies the database file member name, or the diskette or tape file label identifier in the from-file, that is copied. A generic name or *ALL can be specified to copy many database members or diskette file labels, but only a single file label identifier can be copied if the from-file is a tape file. If the from-file is a spooled inline file, FROMMBR(*FIRST) is required.

***FIRST:** The first member (in the order of creation date) in a database from-file is copied. If the from-file is a diskette or tape file, no label identifier is specified by the copy operation when the file is opened. For diskette, a label identifier (LABEL parameter) must be specified in the device file or on an OVRDKTF command. FROMMBR(*FIRST) is required if the from-file is an inline file.

***ALL:** All members of a database from-file, or all file label identifiers for a diskette from-file are copied. FROMMBR(*ALL) is not valid if the from-file is a tape file or inline data file.

If the to-file is a diskette or physical file, the data is copied to like-named to-file members or labels (if TOMBR(*FROMMBR) is specified), or all the from-file members/labels are copied to a single to-file member, diskette label or tape label. If the to-file is a printer file, each member or label is copied to a separate spooled file. If TOFILE(*PRINT) is specified, all the records are copied to a single print output file, and the records for each member or label identifier copied starts on a new print page.

If one of the files copied from a diskette is continued onto another volume, all the files on the continuation volume are also copied (or checked whether they should be copied if a generic name is specified).

member-name: Specify the name of the database from-file member or diskette or tape from-file label identifier of the file member being copied.

generic-member-name:* Specify a generic name to copy all database members that have names with the same prefix, or all diskette data files with the same prefix label identifier. Refer to the description of FROMMBR(*ALL) for more information about copying many from-file members or label identifiers. 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. See generic names for additional information.

TOMBR

Specifies the name of the file member that receives the copied records.

Note:

If the to-file is a printer file, the TOMBR value must be either *FIRST or *FROMMMBR. If a member name or TOMBR(*FROMMMBR) is specified on the CPYF command, or if a to-member name is specified on an override and the member does not exist in the physical to-file, a member is added to the file by the copy operation.

This parameter value is valid for both a single from-file member or label, and for many from-file members or labels (specified by *ALL or a generic name for the FROMMMBR parameter).

Specifying the *FIRST value on the TOMBR parameter is not allowed if the to-file is a physical file that has no members unless either a member name is specified in an override, or CRTFILE(*YES) is specified and the to-file does not already exist. When a physical file is created by the copy operation for the to-file and TOMBR(*FIRST) is specified, the from-file file name is used as the member name in the created file.

***FIRST:** The first member in the database to-file receives the copied records.

***FROMMMBR:** Corresponding from-file and to-file member names or label identifiers are used. This parameter value is valid for both a single from-file member or label, and for many from-file members or labels (specified by *ALL or a generic name for the FROMMMBR parameter). It is ignored if the to-file is a printer or if TOFILE(*PRINT) is specified. If the to-file is a diskette or tape file, or if the to-file is a database file and there is no member override, the TOMBR(*FROMMMBR) value is valid only if the from-file is a database, diskette, or tape file.

member-name: Specify the name of the physical to-file member, or diskette or tape to-file label identifier, to receive the copied records. This parameter value is valid for both a single from-file member or label, and for many from-file members or labels (specified by *ALL or a generic name for the FROMMMBR parameter). If a member with the specified name does not already exist in the physical to-file, the copy operation attempts to add a member with the specified name to the file.

MBROPT

Specifies whether the new records replace or are added to the existing records.

Note:

If the records are being copied to an existing physical file, this parameter must specify *ADD, *UPDADD, or *REPLACE. If the to-file does not exist but CRTFILE(*YES) is specified, the copy operation assumes MBROPT(*ADD) for all records copied to the file after it is created, regardless of the value specified on this parameter.

If *ADD or *UPDADD is specified and the from-file is empty (contains no records), the copy operation completes normally. If *REPLACE is specified and the from-file is empty, the copy operation ends abnormally.

***NONE:** This parameter does not apply to this copy operation. When the to-file is an existing physical file, MBROPT(*NONE) is not allowed; either *ADD, *UPDADD, or *REPLACE must be specified to indicate whether records should be added or replaced in each to-file member that is used.

***ADD:** The system adds the new records to the end of the existing records.

***REPLACE:** The system clears the existing member and adds the new records.

***UPDADD:** The system updates the duplicate key records and adds the new records to the end of the existing records.

Note:

Additional information is available in the File Management topic in the Information Center.

CRTFILE

Specifies, when the CPYF command is used to copy from a physical file or logical file, whether a physical file is created to receive the data if the to-file does not exist. If the to-file already exists when the CPYF command is started, this parameter is ignored. If the to-file is created, the text and public authority of the from-file are used.

***NO:** The to-file must exist when the CPYF command is started. A physical file is not created to receive the data.

***YES:** If the to-file does not exist, a physical file is created with the name specified on the TOFILE parameter. If the from-file is an SQL table, view, or index, that contains a user defined type, datalink, or LOB field type, the physical file created will be an SQL table. In all other instances the to-file created will be a database physical file that is not an SQL table. In addition to the normal copy operation validity checks, the following special conditions must all be true for the copy operation to create a to-file:

- The from-file must be either a physical or logical file.
- A library name must be specified on the TOFILE parameter. The default value, *LIBL, is not allowed.
- There cannot be an override to a different file or library name. The values specified on the CPYF command for the to-file must be used.
- The user running the CPYF command must be authorized to add the file to the TOFILE library, and must have operational authority to the CRTPF command.
- A single record format must be used in the from-file. If the from-file is a logical file with multiple formats, the RCD FMT parameter must specify a record format name.

The members added to the physical file created by the copy operation have the names specified by the TOMBR parameter. If TOMBR(*FIRST) is specified, the to-file member has the same name as the from-file. The MBROPT parameter value is ignored when the to-file is created, and records are added to the new file members.

PRINT Specifies whether copied records, excluded records, or both, are printed. The records are formatted according to the OUTFMT parameter value and written to the IBM-supplied printer file QSYSPRT. QSYSPRT must be spooled (SPOOL(*YES) must be specified in the device file or on the OVRPRTF command) when both copied and excluded records are printed, or when either copied or excluded printouts are requested with TOFILE(*PRINT), because separate printer files are opened for the printouts. If many from-file members or labels are copied, all the members are included in the printer files for copied or excluded records, and each member or label begins on a new print page.

If a selected range of records are requested to be copied using the FROMRCD, TORCD, FROMKEY, TOKEY, or NBRRCD S parameters, then only the records copied or excluded from that range are listed.

If the device file is a printer file, or if PRINT(*EXCLD) or PRINT(*COPIED) is specified, SO-SI characters are not added around the graphic data. OUTFMT(*HEX) can be specified to print the data in hexadecimal format.

***NONE:** No copied, excluded, or error records are printed.

***EXCLD:** Records excluded from the copy operation by the INCCHAR and INCREL parameters are printed.

***COPIED:** Copied records are printed.

***ERROR:** The number of recoverable output error records specified on the ERRLVL parameter are printed.

RCDFMT

Specifies, for a copy operation from a database file only, the name of the record format that is copied. If the from-file is not a logical or physical file, RCDFMT(*ONLY) must be specified.

***ONLY:** The only record format in the from-file is copied. This value is required when the from-file is not a logical or physical file. When the from-file is a logical file, this value is valid only if the file has a single record format.

***ALL:** All record formats in the logical from-file are used. This value is valid for a physical file, and is also valid for a logical file even if the file has only a single record format. If the logical file has many formats, RCDFMT(*ALL) is allowed only if the to-file is a device file or *PRINT, or if the to-file is a physical file and the FMTOPT parameter value is specified as either *NOCHK or *CVTSRC.

record-format-name: Specify the name of the record format copied when the from-file is a logical or physical file. A record format name is optional if the logical file has only a single record format, but either a format name or *ALL must be specified if the from-file has more than one record format. If the logical file is based on more than one physical file member, only the data from the physical file members that are used to derive the specified record format is copied.

FROMRCD

Specifies the record number of the first record in the from-file (or each from-file member) copied. A FROMRCD record number is not valid if a value other than *NONE is specified for the FROMKEY or TOKEY parameters, and is not allowed if the from-file is a keyed logical file.

If the from-file is a physical file or a logical file with an arrival sequence access path, the FROMRCD value is a relative record number that counts both the deleted and nondeleted records that precede it. If the from-file is a device file or inline file, the FROMRCD value is a record number that includes only nondeleted records (even for an I-format diskette file).

If COMPRESS(*YES) is used, and the specified record in a database file member is deleted, the copy operation starts with the first nondeleted record (if any) after the specified record number.

***START:** The copy operation begins with the first record in the file, as determined by the access path or with the first record determined by the FROMKEY parameter value.

starting-record-number: Specify a record number, ranging from 1 through 4294967288, that identifies the first record copied from the from-file. If both FROMRCD and TORCD record number values are specified, the FROMRCD value must be less than or equal to the TORCD value.

TORCD

Specifies the record number of the last record in the from-file (or each from-file member) that is copied. A TORCD record number is not valid if a value other than *NONE is specified for the FROMKEY or TOKEY parameters, or if a value other than *END is specified for the NBRRCDS parameter, or if the from-file is a keyed logical file.

***END:** Records are copied until the end-of-file condition is indicated by the from-file, or until the amount is larger than the TOKEY record key value or the NBRRCDS maximum number of records is reached.

ending-record-number: Specify a record number, ranging from 1 through 4294967288, that identifies the last record copied from the from-file. If both FROMRCD and TORCD record number values are specified, the FROMRCD value must be less than or equal to the TORCD value. If an end-of-file condition is reached before this record number is found, no error messages are issued.

FROMKEY

Specifies, when a file with a keyed access path is copied, that the key value of the first record in the from-file (or each from-file member) is copied. This parameter is valid only when a from-file is a keyed database file, and is not allowed if record number values are specified for the FROMRCD or TORCD parameters.

If no record in the from-file member has a key that is a match with the FROMKEY value, but there is at least one record with a key greater than the specified value, the first record copied is the first record with a key greater than the FROMKEY value. If the specified key value is greater than any record in the member, an error message is sent and the member is not copied.

***NONE:** The first record copied is not selected by key.

Keys With Single-Character String Values:

Specify both of the following elements to identify the key value of the first record copied from the from-file.

Element 1: Number Of Key Fields

number-of-key-fields: Specify the number of key fields used to locate the first record copied. This value must be a number less than or equal to the total number of key fields specified in the data description specification (DDS) for the from-file. If the number is less than the total number of fields in the key, a partial key is used to locate the first record copied.

Element 2: Key String Value

'key-value': Specify a character string (up to 256 characters) that specifies the actual key value for the number of key fields specified by Element 1. The key string value specified must be as long as the total length of all the key fields specified by Element 1, or undesirable results may occur. If the key string specified is shorter than required for the number of key fields used, the string must be padded on the right with zeros. The key string value must be enclosed in apostrophes if it contains blanks or special characters, and it may be specified in hexadecimal format, which is useful if the key contains packed decimal or binary numeric fields, or is a variable-length character field. CCSID conversions are not performed on character fields when a single string is specified.

Keys With A List Of Values:

Specify both of the following elements to identify values for the individual fields of the key, which can be a composite key. This method is generally easier to use if the key contains numeric fields.

Element 1: Build Keys For A List Of Values

***BLDKEY:** Indicates that a list of values is provided for key fields (as opposed to a single-character string value for all fields in the key). *BLDKEY is not valid if any value in the list corresponds to a null-capable field.

Element 2: Key Field Value List

'key-field-value': Specify the list of values (up to 256 characters each) that is applied (in order) to corresponding fields in the from-file key. The maximum number of values (up to 50) allowed in the list is limited to the number of key fields defined in the data description specifications (DDS) for the from-file. If fewer values are provided than the total number of key fields defined for the file, a partial key is used to locate the first record copied.

The values are converted from the displayed form to the type defined in the key field definition. Values specified for character fields are converted from the current job CCSID to the key field CCSID. When a DBCS graphic field is specified, the input string (DBCS data) must be enclosed in SO-SI characters. The SO-SI characters are removed from the input string and the remaining DBCS data is converted from the associated DBCS CCSID of the current job to the DBCS CCSID of the DBCS graphic field. If a value is specified for a character field that is shorter than the actual key field, the value is padded on the right with blanks. A value specified for a numeric key field is converted to the type and precision defined in the DDS for the key field. If a value is either too

large for the corresponding key field or cannot be converted to the type required for the key field, an error message is sent, and the copy operation is not done. For date, time, or timestamp fields, corresponding input values are converted to the format and separator form of the from-file field. For variable-length fields, only specify the character data, not the 2-byte length portion.

TOKEY

Specifies, when a file with a keyed access path is copied, the key value of the last record in the from-file (or each from-file member) copied. This parameter is valid only for a from-file that is a keyed database file, and is not allowed if record number values are specified for the FROMRCD or TORCD parameters, or if a number of records is specified for the NBRRCD parameter.

If there is more than one record in a from-file member with a key that matches the TOKEY value, all those records are copied. If no record in the from-file member has a key that is a match with the TOKEY value, the last record copied is the last one (if any) with a key value less than that of the specified key value.

If there are both ascending and descending fields in the file key, the first (the far left) key field determines whether the copy operation uses an ascending or descending key test to look for the last record to copy.

The user must specify one of the following:

- ***NONE**
- Both elements of **Keys With Single-Character String Values**
- Both elements of **Keys With A List Of Values**

***NONE:** The last record copied is not selected by key.

Keys With Single-Character String Values

Specify the two values that identify the key value of the last record copied from the from-file.

Element 1: Number Of Key Fields

number-of-key-fields: Specify the number of key fields used to locate the last record copied. This value must be a number less than or equal to the total number of key fields specified in the data description specification (DDS) for the from-file. If the number is less than the total number of fields in the key, a partial key is used to locate the first record copied.

Element 2: Key String Value

'key-value': Specify a character string (up to 256 characters) that specifies the actual key value for the number of key fields specified by Element 1. The key string value specified must be as long as the total length of all the key fields specified by Element 1, or undesirable results may occur. If the key string specified is shorter than required for the number of key fields used, the string must be padded on the right with zeros. The key string value must be enclosed in apostrophes if it contains blanks or special characters, and it may be specified in hexadecimal format, which is useful if the key contains packed decimal or binary numeric fields, or is a variable-length character field. CCSID conversions are not performed on character fields when a single string is specified.

Keys With A List Of Values

Specify values for the individual fields of the key, which can be a composite key. This method is generally easier to use if the key contains numeric fields.

Element 1: Build Keys For A List Of Values

***BLDKEY:** Indicates that a list of values is provided for key fields (as opposed to a single-character string value for all fields in the key). *BLDKEY is not valid if any value in the list corresponds to a null-capable field.

Element 2: Key Field Value List

'key-field-value': The list of values (up to 256 characters each) specified is applied (in order) to corresponding fields in the from-file key. The maximum number of values (up to 50) allowed in the list is limited to the number of key fields defined in the DDS for the from-file. If fewer values are provided than the total number of key fields defined for the file, a partial key is used to identify the last record copied.

The values are converted from the displayed form to the type defined in the key field definition. Values specified for character fields are converted from the current job CCSID to the key field CCSID. When a DBCS graphic field is specified, the input string (DBCS data) must be enclosed in SO-SI characters. The SO-SI characters are removed from the input string and the remaining DBCS data is converted from the associated DBCS CCSID of the current job to the DBCS CCSID of the DBCS graphic field. If a value is specified for a character field that is shorter than the actual key field, the value is padded on the right with blanks. A value specified for a numeric key field is converted to the type and precision defined in the DDS for the key field. If a value is either too large for the corresponding key field or cannot be converted to the type required for the key field, an error message is sent, and the copy operation is not done. For date, time, or timestamp fields, corresponding input values are converted to the format and separator form of the from-file field. For variable-length fields, only specify the character data, not the 2-byte length portion.

NBRRCD

Specifies the maximum number of records in the from-file (or each from-file member) copied to the to-file. The records copied start either at the start of the file access path or at the record indicated by the value specified for the FROMRCD or FROMKEY parameter. The TORCD or TOKEY parameters can be used only if NBRRCD(*END) is specified.

This parameter controls the number of records that are copied after the selection value (INCCHAR/INCREL) is applied. It does not control the number of records that are read.

***END:** Records are copied until the end-of-file condition is indicated for the from-file, unless either the TOKEY or TORCD parameter has been specified.

number-of-records: Specify the number of records, ranging from 1 to 4294967288, that are copied to the to-file. Fewer records are copied if an end-of-file condition occurs before the specified number of records have been copied.

INCCHAR

Specifies that records are copied or excluded based on the result of a comparison with a character string value and the data in some position of either a field in the record or the entire record. The comparison done can include searching the record or field for the specified character string value. If INCCHAR is specified for a logical file with many record formats and RCD FMT(*ALL) is specified, the character string is used for selecting records from all the formats.

If both the INCCHAR and INCREL parameters are specified, a record is copied only if it satisfies the requirements for both parameters.

***NONE:** No character string value comparison is used to select which records are copied.

Comparison Values: To specify the comparison that determines which records are copied, four values must be entered. Either *RCD or the name of a field must be entered, followed by the three values that control the comparison: starting position, operator, and character string value. All records that satisfy the relationship specified by the four values are copied to the to-file.

***RCD:** The character string value is compared with the data at the specified starting position in each record copied from the from-file.

***FLD:** This value is the same as the *RCD value.

Element 1: Record Format Field Name

field-name: Specify the name of a field in the record format that is used to make the comparison. The field must be defined as a character field in the data description specification (DDS) for the from-file. When the from-file is a device or inline file, or when the copy operation must process many record formats for a logical from-file (when RCDFMT(*ALL) is specified), a field name cannot be specified (but *RCD is allowed).

Element 2: Field Record Starting Position

starting-position: Specify the position in the field or record where the comparison starts. When a variable-length field name is specified, the position is in the data portion of the variable-length field. For DBCS graphic fields, the position is the DBCS character position. For any operator except *CT, the comparison is done for the length of the character-string value (up to 256 maximum) specified on Element 4 of this parameter. For the *CT operator, the field or record is scanned from the specified starting position to the end of the field or record to determine whether it contains the specified character string. The character string length plus the starting position must not be larger than the length of the field (when a field name is specified) or a record (when *RCD is specified).

Element 3: Operator Value

operator: Specify the operator that indicates the relationship that must exist between the specified portion of the record or field and the character string specified as the last part of the INCCCHAR parameter for the record copied to the to-file. The operators that are used are:

- *EQ** Equal
- *GT** Greater than
- *LT** Less than
- *NE** Not equal
- *GE** Greater than or equal
- *NL** Not less than
- *LE** Less than or equal
- *NG** Not greater than
- *CT** Contains

Element 4: Character String Value

character-string: Specify the character-string (up to 256 characters in length) to be compared with the specified field or record. The character-string length plus the starting position must not be larger than the length of the field (when a field name is specified) or a record (when *RCD is specified).

The character-string value must be enclosed in apostrophes if it contains blanks or special characters, and it can be specified in hexadecimal format. If a field name is specified, the character-string value is converted from the current job CCSID to the field CCSID prior to running the comparison. For variable-length fields, specify only the character data to be compared, not the 2-byte length portion. If a field name is specified, any comparison to a field value that is the null value will test false. For DBCS graphic fields, specify the input string (DBCS data) within SO-SI characters. The SO-SI characters are removed from the input string and the remaining DBCS data is converted from the associated DBCS CCSID of the current job to the DBCS CCSID of the DBCS graphic field.

INCREL

Specifies that records are copied or excluded based on whether certain fields in the record contain data that satisfies specified relationships. As many as 50 field value relations are used to determine whether each record is copied. Include-relationship values are specified only for the INCREL parameter when the from-file is a physical or logical file, and are not valid for a copy operation from all record formats of a logical file with many formats (when RCDFMT(*ALL) is specified).

If both the INCCHAR and INCREL parameters are specified, a record is copied only if it satisfies the requirements for both parameters.

***NONE:** No field value relationships are used to select which records are copied.

Relationship Values: To specify the conditions under which records are copied, a set of values is specified for each condition. Each set must contain exactly four values:

1. One of the logical operators *IF, *AND, or *OR
2. The name of the field compared
3. One of the relational operators (from the list that follows)
4. The comparison value

Values 2 and 4 are compared for the relationship specified by value 3.

The value *IF must be specified as the first value in the first set of comparison values, if there is only one set or several sets of comparison values. If more than one set of comparison values are specified, either *AND or *OR must be specified as the first value in each set after the first set of values. Each INCREL relational set must be enclosed in parentheses.

***IF:** Identifies the first field value relationship that must be satisfied before a record is copied.

***AND:** The field value relational groups on both sides of the *AND value must all be satisfied before a record is copied.

***OR:** If the field value relational group on either side of the *OR value is satisfied, the record is copied.

Element 1: Field Name

field-name: Specify the name of the field compared. The field must exist in the from-file record format, and may be defined as either character or numeric in the DDS for the file.

Element 2: Operator Value

operator: Specify the operator that indicates the relationship that must exist between the field contents in the record and the field value specified as the fourth part of this INCREL relation for this relation to be true. The operators that are used are:

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

***NG** Not greater than

Element 3: Field Value

***NULL:** *NULL can be used as the value to test whether the field value in a record is null. Only the operators *EQ and *NE are allowed if *NULL is specified. An “*EQ *NULL” relation is true only if a field value in a record is null. An “*NE *NULL” relation is true only if a field value in a record is not null.

field-value: Specify the value (up to 256 characters) to be compared with the contents of the specified field. The specified value cannot be another field name. The field value must be enclosed in apostrophes if it contains blanks or special characters, and it may be specified in hexadecimal format. If a CL variable is specified for the value, it must be a character variable. Any non-*NULL comparison to a field value in a record that is null will test false, regardless of the operator used. For variable-length fields, specify only the data portion of the value, not the 2-byte length portion.

Each field value specified is converted from the displayed format to the type defined by the field in the from-file record format. If a value is specified for a character field that is shorter than the actual field, the comparison is performed using only the length of the character string value. A value specified for a character field is converted from the current job CCSID to the CCSID of the from-file field. A value specified for a numeric field is converted to an internal form with the same number of decimal numbers defined in the DDS for the field. For DBCS graphic fields, specify the input string (DBCS data) within SO-SI characters. The SO-SI pair is removed from the input string and the remaining DBCS data is converted from the associated DBCS CCSID of the current job to the DBCS CCSID of the DBCS graphic field. If a value is either too large for the corresponding record format field definition, or cannot be converted to the type required for the field, an error message is sent and the copy operation is not done.

FMTOPT

Specifies, when a physical or logical from-file is copied to a physical to-file, what field-level record format processing (if any) is done. This parameter is ignored if the from-file or to-file is a device or inline file, or if TOFILE(*PRINT) is specified. When either the from-file or to-file is not a database file, records are copied without any field checking and are truncated or padded with blanks or zeros, depending on the characteristics of the to-file.

Note:

Additional information and examples of mapping, truncation, and padding of fields is in the File Management topic in the Information Center and the

Printer Device Programming  book.

When either FMTOPT(*CVTSRC) or FMTOPT(*NOCHK) is specified and the record data copied from any from-file record is not long enough to fill a to-file record, the extra bytes in the to-file record are set to a default value. If a default value was specified in the DDS (DFT keyword) for a field, that field is initialized to the specified default; otherwise, all numeric fields are initialized to zeros, and all character fields are initialized to blanks.

When field-level mapping, dropping, or both are done, any field in the to-file record format that is not set by a corresponding from-file field value (including mapping conversion errors) is set to the default value that was specified on the DFT parameter in the DDS for the file, or to a default value that depends on the field type: blanks for character fields, zeros for numeric fields, current date/time for time/date fields, and null values for null-capable fields.

***NONE:** No field mapping or dropping is done during the copy operation. This value is valid only if the from-file and to-file are not both database files, or if they are both database files and have the same record format. The record formats are the same only if every field exists in both the from-file and to-file formats, and has the same starting position and attributes in both formats. Attributes include whether or not a field is null-capable, CCSID, and the date/time format and separator (if the field is a date/time field). Null values are copied if *NONE is valid and both files are database files.

***NOCHK:** If the record formats of the database files are different, the copy operation continues despite the differences. Record data is copied directly (left to right) from one file to the other. FMTOPT(*NOCHK) is required when copying all record formats from a logical file with multiple formats (when RCDfmt(*ALL) is specified) to a physical file that is of the same type (source or data) as the from-file. If this value is specified, null values are ignored and no conversion of date/time data occurs and no CCSID conversions are done.

***CVTSRC:** This value is used to copy between database files, from a source file to a data file, or from a data file to a source file. It is valid only when the from-file and to-file are different types (source and data). The file type conversion is done as follows:

- If the to-file is a data file, the from-file sequence number and date fields are dropped, and the source data part of each from-file record is copied to the to-file, as described for FMTOPT(*NOCHK).
- If the to-file is a source file, sequence number and date fields are appended, and the from-file record data is copied to the source data part of each to-file record, as described for FMTOPT(*NOCHK). The SRCOPT and SRCSEQ parameters are used to control the sequence numbers created in the to-file. Null values are ignored and no conversion of date/time data occurs.

Note:

When either the from-file or the to-file is not a database file, FMTOPT(*CVTSRC) is not required for copying from a source file to a data file or from a data file to a source file. Sequence number and date fields are appended or dropped automatically, depending on the file types. If the to-file is a source physical file, the SRCOPT and SRCSEQ parameters can be used to control the sequence numbers created for records copied to the to-file.

***MAP:** Fields with the same name in the from-file and to-file record formats are copied, and any fields in the to-file that do not exist in the from-file format are set to one of the following:

- The default value indicated by the DFT value in the data description specification (DDS) for the to-file.
- Blanks for character fields and zeros for numeric fields.
- Current date/time for date/time fields.
- The null value for null-capable fields.

If *MAP is specified, *DROP can also be specified. Mapped fields may have different starting positions in the from-file and to-file record formats.

If *MAP is specified and a valid conversion is defined between the from-file field CCSID and the to-file field CCSID, the character data is converted to the to-file field CCSID.

If the from-file field CCSID or the to-file field CCSID is 65535, the character data is not converted.

*MAP allows conversion of date/time data and the copying of null values.

***DROP:** This value must be specified for field-level mapping if any of the field names in the from-file record format do not exist in the to-file format. If *DROP is specified, *MAP can also be specified. When *DROP is specified, all the field names that exist in both record formats must have the same attributes and relative positions in the from-file and to-file record formats, or *MAP must also be specified. Null values are copied if *DROP is valid.

***CVTFLOAT:** This value is specified when CPYF processes each floating point field identified by the external description of the output database physical file and convert it from System/370 hexadecimal format to the IEEE format used by iSeries 400 computers.

***NULLFLAGS:** This value is specified when CPYF takes the byte following each field identified as being nullable by the external description of the output file, and use it as a flag to indicate if the corresponding input field is null. If the byte is blank ('40'X) or contains '00'X, the data is considered to be not null. Any other value for the flag causes the corresponding input field data to be ignored and the output value set to null.

Note:

If *CVTFLOAT or *NULLFLAGS is specified and the input file is externally described, the input file external description will not be used in doing the mapping of the copied data. If *CVTFLOAT or *NULLFLAGS is specified, any other value is ignored (unless both are specified). TOFILE must be an externally described physical data file. The following parameter values cannot be specified when *CVTFLOAT or *NULLFLAGS is specified:

- RCD_FMT(*ALL) when the from-file is a multiple format logical file
- A value other than default for CRTFILE (unless the TOFILE already exists causing *YES to be ignored), FROMKEY, TOKEY, INCCHAR, INCREL, SRCOPT, and SRCSEQ

Attention:

*CVTFLOAT and *NULLFLAGS must only be used for conversion of data to iSeries 400 format, and they must be used correctly to avoid possible data corruption.

Note:

Additional information is available in the File Management topic in the Information Center.

SRCOPT

Specifies, only for copying to a source physical to-file, whether new sequence numbers are inserted in the sequence number fields, and whether the date fields are set to zero. If *SEQNBR is specified, the SRCSEQ parameter gives the values that control the new sequence numbers created in the records copied to each to-file member.

***SAME:** The value does not change.

***SEQNBR:** New source sequence numbers are inserted in the records copied to the to-file. The new sequence numbers are controlled by the SRCSEQ parameter value. This value is valid only if the to-file is a source physical file. If *SEQNBR is specified, *DATE can also be specified.

***DATE:** The source date field is set to zero in the records copied to the to-file. This value is valid only if the to-file is a source physical file. If *DATE is specified, *SEQNBR can also be specified.

SRCSEQ

Specifies, only when SRCOPT(*SEQNBR) is also specified, the sequence number that is given to the first record copied to the to-file, and what increment value is used to renumber all other records copied. This parameter allows the copied file to have as many as 999 999 records with

unique sequence numbers. If SRCOPT(*SEQNBR) is specified, but SRCSEQ is not specified, SRCSEQ(1.00 1.00) is assumed; the copied records are renumbered sequentially beginning with 1.00, and the whole number increment of 1 is used.

If SRCOPT(*SEQNBR) is specified and the maximum sequence number value of 9999.99 is reached, then all remaining records copied to the to-file member also have a sequence number value of 9999.99.

Element 1: Starting Value

1.00: The first source record copied to the to-file has a sequence number of 0001.00.

starting-value: Specify a value ranging from 0000.01 through 9999.99 that is the sequence number of the first source record copied to the to-file. A whole number of no more than four digits and/or a fraction of no more than 2 digits is specified. If the starting value contains a fraction, a decimal point must be used. Examples are .01 and 3250.4. (If a value has a fraction of .00, such as 5000.00, it is specified without the fraction; either 5000 or 5000.00 is valid.)

Element 2: Increment Value

1.00: The copied source records are renumbered in the to-file with whole number increments of 1. (1.00, 2.00, 3.00, ...)

increment-value: Specify a value ranging from 0000.01 through 9999.99 that is used as the increment value for renumbering all source records copied after the first record. A whole number of no more than four numbers and/or a fraction of no more than two numbers can be specified. If the increment value contains a fraction, a decimal point must be used. For example, if SRCSEQ(5000 10) is specified, the first record copied to the file is numbered 5000.00, the second is 5010.00, the third is 5020.00, and so forth. If SRCSEQ(1 .25) is specified, the copied records are numbered 1.00, 1.25, 1.50, 1.75, 2.00, and so forth.

OUTFMT

Specifies, if TOFILE(*PRINT) is specified, whether the copied records are printed in character or hexadecimal format.

Note:

This parameter is used only when TOFILE(*PRINT) is specified or the PRINT parameter specifies *EXCLD, *COPIED, or both. This parameter has no effect on the printed output when the to-file is a program-described printer file. The OUTFMT parameter is ignored when the copy operation does not need to produce a formatted printer file.

***CHAR:** Records are printed in character format only.

***HEX:** Records are printed in both character format and hexadecimal format.

ERRLVL

Specifies the maximum number of recoverable read or write errors for the file that are tolerated during the copy operation for a single database from-file member or tape from-file label identifier. This parameter is ignored if the from-file is not a physical, logical, or tape file and the to-file is not a physical file.

0: If any recoverable error occurs, the copy operation ends at the file member in which the error occurs.

***NOMAX:** No maximum number of errors is specified, and all recoverable errors are tolerated. The copy operation continues regardless of the number of recoverable errors found.

number-of-errors: Specify a value that specifies the maximum number of recoverable errors that is allowed in each from-file member or label that is copied. If one more error occurs than the value specified here, the copy operation is ends.

COMPRESS

Specifies whether the to-file contains a compressed form of the from-file. Compression occurs when deleted records in the from-file are not copied to the to-file. COMPRESS(*NO) is used to copy both deleted and nondeleted records only when the from-file and to-file are both physical files. If MBROPT(*ADD) is specified with COMPRESS(*YES), deleted records that existed in the to-file member before the copy operation are not compressed.

***YES:** The records copied to the to-file are compressed. Deleted records that exist in the from-file are not copied to the to-file. Only nondeleted records are copied, and they are renumbered consecutively in the to-file. That is, the relative record numbers of all nondeleted records that occur after the first deleted record in the from-file are different in the to-file. No physical record data, such as source sequence numbers, is changed by the copy operation as a result of specifying COMPRESS(*YES). If from-file is delete-capable and the to-file is nondelete-capable, COMPRESS(*YES) must be specified.

***NO:** Both the deleted and nondeleted records are copied to the to-file. If the from-file is a database file that is copied in arrival sequence, the relative record numbers in the from-file are not changed in the to-file if MBROPT(*REPLACE) is also specified. If the from-file is a database file that is copied in keyed sequence (no deleted records are contained in an access path), COMPRESS(*NO) is ignored.

Note:

If the from-file is a keyed physical file and a record number value is specified for either the FROMRCD or TORCD parameters, the from-file is copied in arrival sequence; therefore, the COMPRESS parameter determines whether deleted records are copied. If COMPRESS(*NO) is specified, the default value must be used for the INCCHAR and INCREL parameters.

Table 1. Valid Parameters for Copying from or to Database Files (CPYF Command)

Parameter	Physical ¹		Logical ¹ From ²
	From	To	
FROMFILE	X		X
TOFILE		X	
FROMMBR	X		X
TOMBR		X	
MBROPT		X	
CRTFILE	X ³	X ³	X ³
PRINT	X ⁴	X ⁴	X ⁴
RCDFMT			X
FROMRCD	X		X ⁵
TORCD	X		X ⁵
FROMKEY	X		X
TOKEY	X		X
NBRRCDS	X		X
INCCHAR	X		X
INCREL	X		X
FMTOPT	X	X	X
SRCOPT		X	
SRCSEQ		X	
OUTFMT	X ⁴	X ⁴	X ⁴
ERRLVL	X	X	X

COMPRESS

X⁶

X⁶

Note:

- 1 DDM files can be either physical or logical files.
- 2 Cannot copy to a logical file.
- 3 If the to-file does not exist before the copy operation begins, and the from-file is either a physical or logical file, CRTFILE(*YES) can be specified to allow the copy operation to create a physical file for the to-file.
- 4 A program-described printer file can be specified for the to-file to create a printout that has no special formatting or page headings, or TOFILE(*PRINT) can be used to create a formatted printout. PRINT(*EXCLD) or PRINT(*COPIED) may be specified to create a formatted printout for any copy operation. When a printout is requested (by either TOFILE(*PRINT) or by the PRINT parameter), the UTFMT parameter specifies whether the data is printed in character form or both character and hexadecimal form.
- 5 The FROMRCD and TORCD parameter values are valid for a from-file that is a logical file if it has only an arrival sequence access path (no keyed access path).
- 6 Cannot copy from a delete capable file to a file that is not delete capable unless COMPRESS(*YES) is specified.

Table 2. Valid Parameters for Copying from Device Files (CPYF Command)

Parameter	Diskette	Tape	Inline Data
FROMFILE	X ¹	X	X
FROMMBR	X	X	
PRINT	X ²	X ²	X ²
FROMRCD	X	X	X
TORCD	X	X	X
NBRRCDS	X	X	X
INCCHAR	X	X	X
UTFMT	X ²	X ²	X ²
ERRLVL		X	

Note:

- 1 If both the from-file and to-file are diskette files, the to-file must be spooled (SPOOL(*YES) is specified on CRTDKTF, CHGDKTF, or OVRDKTF command).
- 2 A program-described printer file can be specified for the to-file to create a printout that has no special formatting or page headings, or TOFILE(*PRINT) can be used to create a formatted printout. PRINT(*EXCLD) or PRINT(*COPIED) may be specified to create a formatted printout for any copy operation. When a printout is requested (by either TOFILE(*PRINT) or by the PRINT parameter), the UTFMT parameter specifies whether the data is printed in character form or both character and hexadecimal form.

Table 3. Valid Parameters for Copying to Device Files (CPYF Command)

Parameter	Diskette	Tape	Printer ¹
TOFILE	X ²	X	X
TOMBR	X	X	
PRINT	X ¹	X ¹	X ¹
UTFMT	X ¹	X ¹	X ¹

Parameter	Diskette	Tape	Printer ¹
-----------	----------	------	----------------------

- Note:**
- ¹ A program-described printer file can be specified for the to-file to create a printout that has no special formatting or page headings, or TOFILE(*PRINT) can be used to create a formatted printout. PRINT(*EXCLD) or PRINT(*COPIED) may be specified to create a formatted printout for any copy operation. When a printout is requested (by either TOFILE(*PRINT) or by the PRINT parameter), the OUTFMT parameter specifies whether the data is printed in character form or both character and hexadecimal form.
 - ² If both the from-file and to-file are diskette files, the to-file must be spooled (SPOOL(*YES) is specified on CRTDKTF, CHGDKTF, or OVRDKTF command).

Examples for CPYF

The following examples of the CPYF command show the type(s) of files that are copied, and the function provided by various parameters.

Example 1: Physical File to Physical File

```
CPYF FROMFILE(PERSONNEL/PAYROLL)
     TOFILE(TESTPAY/PAYROLL) MBROPT(*ADD)
     CRTFILE(*YES) ERRLLVL(10)
```

This command copies all of the records in the physical file named PAYROLL in the PERSONNEL library to the file PAYROLL in the TESTPAY library. If the from-file contains more than one member, only the first member is copied. If TESTPAY/PAYROLL does not exist, it is created before the records are copied and a member with the same name as the from-file is added to TESTPAY/PAYROLL to receive the copied records.

Because MBROPT(*ADD) is specified, the copied records are added to any existing records in the to-file member. Because RCDfmt(*NONE) is assumed, the to-file TESTPAY/PAYROLL must have the same record format as the from-file. If the to-file (TESTPAY/PAYROLL) is created by the copy operation, it will have the same record format and access path as the from-file (PERSONNEL/PAYROLL). If more than ten recoverable errors occur during the copy operation, the operation ends.

If FROMMBR(*ALL) and TOMBR(*FROMMBR) had also been specified, all of the members in the from-file would be copied to corresponding members (having the same names) in the to-file. For each from-member that has no corresponding to-member, a member is added to the to-file and all the records in the from-member are copied to the new member. For each to-member that already exists, only new records are added to the member. No updates are made to existing records on any type of copy operation. If the to-file contains members for which there are no corresponding members in the from-file, the to-file contains more members than the from-file after the copy operation.

If more than ten recoverable errors occur within a member being copied, the copy operation ends at that point, and remaining members are not copied. ERRLLVL(*NOMAX) can be specified to tolerate all recoverable errors, so the copy operation does not end no matter how many recoverable errors occur in a particular file member.

Example 2: Physical File to Physical File

```
CPYF FROMFILE(PERSONNEL/EMP1)
     TOFILE(PERSONNEL/VACLEFT)
     FROMMBR(VAC) MBROPT(*REPLACE)
     FROMKEY(1 X'0008872F') TOKEY(1 X'0810199F')
     INCREL((*IF VAC *GT 5.0)) FMOPT(*MAP *DROP)
```

In this example, the to-file (VACLEFT) is an existing physical file, but its record format differs from that of the physical file named EMP1, which is being copied. Both files are in the PERSONNEL library. The from-file contains employee records and has a key (employee number). The records selected in the

from-file are those with employee numbers ranging from 008872 through 810199. Only records for employees with more than five days of vacation (VAC) are mapped to the receiving file. Records are selected from member VAC, and they replace existing records in the first member of file VACLEFT.

Because the key for the file is a packed decimal number, the FROMKEY and TOKEY values must be specified as hexadecimal strings, and the leading zeros and hexadecimal sign are required in the value. An alternative way of specifying the same key value range follows:

```
FROMKEY(*BLDKEY 8872) TOKEY(*BLDKEY 810199)
```

When *BLDKEY is specified, the copy operation converts each number to the format required for the file key definition. Because only a single value is specified, only one key field is used. The *BLDKEY form of the FROMKEY and TOKEY parameters allows omission of leading zeros and a positive sign value when the key is numeric.

If the key for a file is a composite of more than one key field, the *BLDKEY form is used with a list of values for the FROMKEY and TOKEY parameters. For instance, if the key fields for a file are a sales region (10 characters) and the sales for the last month (7 packed decimal numbers with 2 decimal positions), a complete key is specified in either of the following ways:

```
FROMKEY(*BLDKEY (GEORGIA 99.50))
```

```
FROMKEY(2 X'C7C5D6D9C7C9C14040400009950F')
```

When the *BLDKEY form is used, each character field is padded with blanks, and each numeric field is converted to the actual key format with the value shifted left or right to correctly align the decimal point.

Example 3: Physical Data File to Physical Source File

```
CPYF FROMFILE(MYLIB/DATAFILE) TOFILE(QIDU/QTXTSRC)
FROMMMBR(A1) TOMBR(*FROMMMBR)
MBROPT(*REPLACE) FMTOPT(*CVTSRC)
```

This command copies records from physical file DATAFILE in library MYLIB, which is defined as FILETYPE(*DATA), to physical file QTXTSRC in library QIDU, which is defined as FILETYPE(*SRC). Because the two database files are of different types, FMTOPT(*CVTSRC) must be specified. Records are copied to member A1, which has the same name as the from-file member. Values are assigned to the sequence number source field of the records copied to the source file, starting with 1.00 and incremented by 1.00. If SRCOPT(*SEQNBR) is specified, the SRCSEQ parameter is used to control the sequence numbers that are created. The date source field is always set to zeros.

Example 4: Logical File to Physical File

```
CPYF FROMFILE(DEPTS/SALES) TOFILE(DEPTS/YTDSALES)
FROMMMBR(TOTSALES) TOMBR(MARCH) RCDfmt(AA)
NBRRCDs(5) MBROPT(*REPLACE)
```

This command copies five records from member TOTSALES of logical file SALES (in library DEPTS) to member MARCH in the physical file YTDSALES (in library DEPTS). If member MARCH does not exist, it is created and added to the to-file automatically by the copy operation. Only records from the logical file SALES in library DEPTS that use record format AA are copied, and they are copied to YTDSALES, which has the same format. After the copy operation, the MARCH member contains only five nondeleted records, because all records in that member are first cleared, then only the data in the first five records (in keyed sequence) in the TOTSALES member are copied to it.

Example 5: Device File to a Physical File

```
CPYF FROMFILE(QDKT) TOFILE(QGPL/QCLSRC)
FROMMMBR(PAY*) TOMBR(*FROMMMBR)
MBROPT(*REPLACE) SRCOPT(*SEQNBR)
SRCSEQ(1 .25)
```

This command copies records from the generic set of diskette labels with names that start with the characters PAY. They are copied to like-named members in source file QCLSRC in the QGPL library. Even though the to-file is a source file, a diskette file (QDKT) defined as FILETYPE(*DATA) is used as the from-file, because QDKT is more efficient than a device file defined as FILETYPE(*SRC). For each label copied, the sequence number of the first record is 1.00 and is incremented by .25 for each subsequent record. The source date field is automatically set to zeros.

Example 6: Physical File to the Printer

```
CPYF FROMFILE(TEMPFILE) TOFILE(*PRINT)
    FROMMBR(EMP1) FROMKEY(1 448762)
    NBRRCD(20) OUTFMT(*HEX)
```

This command copies records from member EMP1 in the file named TEMPFILE. The records are employee records. One key field, the employee number, is used to search the record keys. Twenty records, starting with employee number 448762, are copied to the IBM-supplied printer file QSYSPRT and listed in both character and hexadecimal format. The IBM-supplied printer file is indicated by coding TOFILE(*PRINT).

Example 7: Physical File to a Device File

```
CPYF FROMFILE(PERSONNEL/PAYROLL)
    TOFILE(DISK1) FROMMBR(VAC1)
    INCCHAR(NAME 1 *CT SMITH)
    INCREL((*IF VAC *GT 10.5)(*AND HOLIDAYS *EQ 0))
```

This command copies all employee records of employees whose last name is SMITH and that have accumulated more than ten and a half vacation days, none of which is holidays, from the PAYROLL file in the PERSONNEL library to a diskette. The file member name copied is VAC1. The vacation (VAC) and holiday (HOLIDAYS) fields are defined as packed decimal, but a value is specified in character form on the INCREL parameter. The diskette device file used is DISK1, which contains the label of the file being copied to, and other diskette attributes such as location and volume ID.

Example 8: Physical File to Device Files

```
CPYF FROMFILE(PERSONNEL/PAYROLL) TOFILE(DISK1)
    FROMMBR(*ALL) TOMBR(*FROMMBR)
```

This command copies all members of file PAYROLL in the PERSONNEL library to data files on diskette (device file DISK1). Each from-file member name must be a valid diskette label identifier; if not, use the RNMM (Rename Member) command to rename the members in the from-file before they are copied.

Error messages for CPYF

*ESCAPE Messages

CPF2807

Cancel reply received for message &7.

CPF2816

File &1 in &2 not copied because of error.

CPF2817

Copy command ended because of error.

CPF2818

*FROMMBR value is not allowed on TOMBR parameter.

CPF2835

INCCHAR starting position and length too long.

CPF2857
Multiple member or label copy not allowed with override.

CPF2858
File attributes not valid for printed output.

CPF2859
Shared open data path not allowed.

CPF2864
Not authorized to file &1 in library &2.

CPF2875
Wrong file member or label opened.

CPF2883
Error creating file &1 in library &2.

CPF2888
Member &3 not added to file because of error.

CPF2904
Diskette labels not valid for multiple label copy.

CPF2906
Value not valid for INCREL field.

CPF2909
Error clearing member &3 in file &1 in &2.

CPF2949
Error closing member &3 in file &1 in &2.

CPF2952
Error opening file &1 in library &2.

CPF2968
Position error occurred copying file &1 in &2.

CPF2971
Error reading member &3 in file &1.

CPF2972
Error writing to member &3 in file &1.

CPF2975
Error while reading from keyed file.

CPF2976
Number of errors greater than ERRLVL value.

CPF3140
Initialize or copy of member &2 canceled.

CPF3143
Increments not allowed for member &2.

CPF3148
New records need too much space for member &2.

CPF3150
Data base copy failed for member &2.

CPF9212
Cannot load or unload DDM file &2 in &3.

CPYFRMDIR (Copy From Directory) Command Description

CPYFRMDIR Command syntax diagram

Purpose

The Copy From Directory (CPYFRMDIR) command is used to copy directory data from the local system to a tape or diskette device. This directory data can then be copied to other remote systems by using the Copy To Directory (CPYTODIR) command on the remote systems. This function allows the remote system to begin a directory shadowing environment with the local system by shadowing changes made to the directory data from the local system.

Caution: Do not use this command as a backup utility to save and restore directory data for data recovery purposes. Follow the normal backup and recovery procedure guidelines described in the Backup and Recovery topic in the Information Center.

Restriction: The user must have security administrator (*SECADM) authority to use this command.

Required Parameters

LABEL

Specifies the name that identifies the device file label on the tape or diskette to be copied. A maximum of 17 characters can be specified for tape devices and 8 characters for diskette devices.

DEV Specifies the names of the tape or diskette devices used for the copy operation. Each tape or diskette device name must already be known on the system by a device description.

diskette-device-name: Specify the name of the diskette device to be used for the copy operation.

tape-device-name: Specify the names of one or more tape devices used for the copy operation. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume.

Optional Parameters

SYSNAME

Specifies the names of the remote systems that copies the directory data from the tapes or diskettes created by this command. The names of the remote systems on this parameter are added to the list of system names that are collecting changes to directory data from the local system.

Note: You must include the names of all the remote systems that use the tapes or diskettes created by this command to ensure that all changes to directory data are sent to the remote systems during a normal shadowing session.

VOL Specifies one or more volume identifiers used by the file. More information on this parameter is in Commonly used parameters.

***NONE:** No volume identifiers are specified for the file. No volume identifiers are checked.

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are placed in a device.

SEQNBR

Specifies the sequence number of the data file on the tape being processed. The four-position file sequence number is read from the first header label of the data file.

***END:** The copy operation begins after the last sequence number on the tape volume.

file-sequence-number: Specify the sequence number of the file that is used. Valid values range from 1 through 9999. Valid values range from 0001 through 9999.

ENDOPT

Specifies the operation that is automatically performed on the tape volume after the operation ends. If more than one volume is included, this parameter applies only to the last tape volume used; all other tape volumes are rewound and unloaded when the end of the tape is reached.

***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.

EXPDATE

Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

***PERM:** The data file is permanently protected. An expiration date of 999999 is assigned.

expiration-date: Specify the date when protection for the file ends.

Example for CPYFRMDIR

```
CPYFRMDIR  DEV(TAP01)  SYSNAME(CHICAGO NEWYORK)
```

This command copies all of the directory data from the local system to tape device TAP01. CHICAGO and NEWYORK are added to the list of systems that collect changes to the directory data from the local system.

Error messages for CPYFRMDIR

***ESCAPE Messages**

CPF90A8

*SECADM special authority required to do requested operation.

CPF90FB

Directory data not copied because of errors.

CPYFRMDKT (Copy From Diskette) Command Description

CPYFRMDKT Command syntax diagram

Purpose

The Copy from Diskette (CPYFRMDKT) command copies one or more data files from diskette to an output file or to the printer. The from-file must be a diskette file for this command, but the to-file can be a physical, DDM, program-described printer, tape, or diskette file, or *PRINT to print the records using the IBM-supplied printer file QSYSPRT.

Note:

For more information on DDM files, see the Distributed Data Management topic in the Information Center.

This command offers a subset of the parameters available on the Copy File (CPYF) command. If you need parameters that are not available on the CPYFRMDKT command, you can either use overrides for the from-file or to-file, or use the CPYF command or a combination of file overrides with the Copy File (CPYF) command.

One label, a generic set of labels, or all labels from the diskette are copied. The File Management topic in the Information Center has a complete description of the combinations allowed and how to specify them.

The to-file must exist when the CPYFRMDKT command is started. This command does not create the to-file, but it *does* add a member to an existing physical file if the member does not already exist in the to-file.

Note: This command cannot be used to copy save/restore type files.

Restriction: A file's open data path (ODP) cannot be shared with any other program in the job (routing step) during the copy operation.

Required Parameters

FROMFILE

Specifies the qualified name of the diskette device file that contains the copied records.

The name of the diskette 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.

diskette-file-name: Specify the name of the diskette device file that contains the copied records.

TOFILE

Specifies the qualified name of the file that receives the copied records.

Note: If no library qualifier is given, *LIBL is used to find the file. The device file can be a diskette, tape, or program-described printer file. If a diskette file is used for the TOFILE parameter, the diskette spool writer must not be active and the diskette file must be defined with SPOOL(*YES), because the system is not able to allocate the same diskette device for both the from-file and the to-file to do the copy operation.

***PRINT:** The records are copied to the IBM-supplied printer file QSYPRT, and the file is formatted according to the OUTFMT parameter. The IBM-supplied printer file QSYPRT may not be overridden to a different file name, and it must have the RPLUNPRT(*YES) and CTLCHAR(*NONE) attributes.

The name of the to-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.

to-file-name: Specify the name of the physical file or device file that receives the copied records. If no library qualifier is given, *LIBL is used to find the file.

Optional Parameters

FROMLABEL

Specifies the label identifier of a single-diskette data file or the generic identifier for a group of diskette data files that are copied, or indicates that all data files are copied from the diskette.

***DKTF:** Specifies that the data file label identifier in the diskette device file description is used to identify the file on the diskette that is copied (it can also be specified in an override for the from-file).

***ALL:** All data files on the diskette volumes are copied. If the to-file is either a spooled diskette or database physical file, the records can be copied to corresponding diskette labels or physical file members in the to-file with the same name (by specifying TOMBR(*FROMLABEL)), or it can be copied to a single label or member that contains a concatenation of all records from all data files copied from a diskette. If the to-file is a printer file, each data file is copied to a separate spooled file. If TOFILE(*PRINT) is specified, all the data files on the diskette are copied to a single-print output file and the records for each data file that is copied begins on a new print page.

If FROMLABEL(*ALL) is specified and a LABEL parameter value is also specified on an Override Diskette File (OVRDKTF) command, only the single-file label identifier specified in the override is copied.

data-file-identifier: Specify the label identifier of the data file that is read from the diskette. If a different LABEL parameter value has been specified on an OVRDKTF command, the label identifier specified on the OVRDKTF command is used instead of the value specified on this parameter.

generic-data-file-identifier:* Specify the generic label identifier of the data files that are copied from the diskette. 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. See generic names for additional information.

If a generic identifier is specified for the FROMLABEL parameter and a LABEL parameter value is also specified on an OVRDKTF command, only the single-file label identifier specified on the override is copied.

If one of the files being copied from a diskette is continued onto another volume and FROMLABEL(*ALL) or a generic label identifier is specified, all the files on the continuation volume are processed. The system attempts to copy files from all diskettes until it completes processing for a diskette with no copied file that continues onto another volume.

TOMBR

Specifies the name of the file member that receives the copied records.

Note:

If the TOFILE is a diskette or tape device file, TOMBR specifies the label identifier of the data file to which the records are copied. If the TOFILE is a printer file or *PRINT, then *FROMLABEL or *FIRST must be specified. A physical file member is added with the name specified by this parameter (including a name implied by *FROMLABEL) if one does not exist.

***FROMLABEL:** Specifies that all files which are identified by the data file identifier specified in the FROMFILE parameter, are copied to corresponding members or diskette or tape labels in the physical to-file. If a member or file with a corresponding name does not exist in the to-file, then a member or a file with that name is added to a physical to-file.

If a single data file identifier was specified as a value for the FROMLABEL parameter, then a member in the to-file with the same name receives the copied records. If a generic data file identifier or *ALL was specified as a value for the FROMLABEL parameter, each file label in the from-file is copied to a corresponding member or label in the to-file. If the to-file is a tape file and *FROMLABEL is specified, then a single-data file identifier or *DKTF must be specified for the FROMLABEL parameter. If the to-file is a tape or diskette device file, the label in the device file description is used.

***FIRST:** The first member in the physical file receives the copied records.

to-member-name: Specify the name of the physical file member or the file label identifier of the diskette or tape data file that receives the copied records. If the label identifier for the tape file is more than 10 characters long or contains special characters, then it must be specified on the Create Tape File (CRTTAPF), Change Tape File (CHGTAPF), or Override Tape File (OVRTAPF) command.

FROMDEV

Specifies the name of diskette devices from which the diskette from-file is copied.

***DKTF:** The value specified in the diskette device file is used to indicate the devices used.

device-name: Specify the names of diskette devices used when copying records from the from-file. The order in which the device names are specified is the order in which the tapes on the devices are read. Each device name must already be known on the system by a device description.

FROMVOL

Specifies the diskette that is used.

***DKTF:** The diskette volume identifiers in the diskette device file are used to identify the diskette file that is copied (it can also be specified in an override for the from-file).

***NONE:** No volume identifier checking is done.

volume-identifier: Specify up to 50 volume identifiers used to identify the diskettes that are copied. Each identifier can have 6 alphanumeric characters or less.

MBROPT

Specifies whether the new records replace or are added to the existing records.

Note:

If the to-file is a device file, this parameter is ignored. If the to-file is a physical file, this parameter is required.

***NONE:** No records are added or replaced in a member. This value is valid only for a copy to a device file.

***ADD:** The system adds the new records to the end of the existing records.

***REPLACE:** The system clears the existing member and adds the new records.

NBRRCD

Specifies the number of records copied to the to-file.

***END:** Records are copied until the end-of-file condition is indicated.

number-of-records: Specify the number of records, ranging from 1 to 4294967288, that are copied to the to-file. Fewer records are copied if an end-of-file condition occurs before the specified number of records have been copied.

OUTFMT

Specifies, if TOFILE(*PRINT) is specified, whether the copied records are printed in character or hexadecimal format.

***CHAR:** Records are printed in character format only.

***HEX:** Records are printed in both character format and hexadecimal format.

Examples for CPYFRMDKT

Example 1: Copying Records to a Database File

```
CPYFRMDKT FROMFILE(QDKT) TOFILE(MASTER/PAYROLL)
          FROMLABEL(MONTH1) MBROPT(*REPLACE)
```

This command copies records from a diskette using the diskette device file QDKT. The diskette device specified on the QDKT file description is created. The data file on the diskette that is copied is identified by label MONTH1. The records are copied to the physical database file PAYROLL in library MASTER and replaces the existing records in member MONTH1 (which is implied by the parameter default of TOMBR(*FROMLABEL)).

Example 2: Printing Copied Records

```
CPYFRMDKT FROMFILE(QDKT) TOFILE(*PRINT)
          FROMDEV(DKT2) FROMLABEL(MONTH*) FROMVOL(PAY1)
```

This command copies from a diskette, by using diskette device file QDKT and the diskette device DKT2, the generic set of labels that start with the characters MONTH. The diskette volume identifier is specified on the command, which eliminates the need for a separate override command. The records are listed on the printer by using IBM-supplied printer file QSYSPRT and printed in character format, which is the default for the OUTFMT parameter.

Error messages for CPYFRMDKT

***ESCAPE Messages**

CPF2816

File &1 in &2 not copied because of error.

CPF2817

Copy command ended because of error.

CPF2818

*FROMMBR value is not allowed on TOMBR parameter.

CPF2857

Multiple member or label copy not allowed with override.

CPF2858

File attributes not valid for printed output.

CPF2859

Shared open data path not allowed.

CPF2875

Wrong file member or label opened.

CPF2888

Member &3 not added to file because of error.

CPF2904

Diskette labels not valid for multiple label copy.

CPF2909

Error clearing member &3 in file &1 in &2.

CPF2949

Error closing member &3 in file &1 in &2.

CPF2952

Error opening file &1 in library &2.

CPF2971

Error reading member &3 in file &1.

CPF2972

Error writing to member &3 in file &1.

CPF9212

Cannot load or unload DDM file &2 in &3.

CPYFRMIMPF (Copy From Import File) Command Description

CPYFRMIMPF Command syntax diagram

Purpose

The Copy From Import File (CPYFRMIMPF) command copies all or part of an *import file* to the TOFILE. The term *import file* is used to describe a file created for purposes of copying data between heterogeneous databases. The import file (FROMSTMF or FROMFILE parameter) is called the *from file* for this command.

An important aspect of this command is its ability to copy the data in parallel. By using the Change Query Attributes (CHGQRYA) command, the number of tasks used to perform the copy is determined by the DEGREE parameter of the CHGQRYA command. The system feature DB2 Symmetric Multiprocessing for OS/400 must be installed for using multiple tasks. See the CHGQRYA command and the example section.

Some of the specific functions that can be performed by the CPYFRMIMPF command include the following:

- Copying a from file to an externally described physical file. The TOFILE must exist on the system before the copy can occur.
 - Limiting the range of records copied based on starting and ending relative record numbers.
- Adding records to an existing file member or replace the contents of a receiving file member (MBROPT parameter).

Error Handling: The escape message CPF2817 is sent for many different error conditions that can occur during a copy operation. At least one diagnostic message that indicates the specific error condition always comes before the escape message. More information on handling errors is in the File Management topic in the Information Center.

Overrides: Overrides are processed for all files.

Status Message: During the running of the CPYFRMIMPF command, message CPI2801 is sent as a status message informing the interactive user that a copy is occurring. More information on preventing status messages from appearing is in the File Management topic in the Information Center.

Restrictions:

1. The from file and TOFILE cannot be the same file.
2. The TOFILE must exist prior to the copy.
3. The TOFILE will not have the same relative record numbers as the from file.
4. The from file must be a source file, or a valid file with 1 field that is not a numeric data type.
5. If the from file is defined with the SHARE(*YES) option for the file, unpredictable results can occur. Therefore, if the file is defined with SHARE(*YES), the user should make sure the file is not opened by any process prior to the copy.

Performance:

To increase the performance of the copy:

1. Delete any logical keyed files based on the TOFILE.
2. Remove all constraints and triggers of the TOFILE.
3. Ensure the FROMFILE records will be copied correctly by attempting to copy a few of the records, by using the FROMRCD and number of records option, before copying all the records.
4. Use the ERRlvl(*NOMAX) parameter after knowing the data can be copied correctly.

Notes For Delimited Data:

1. A delimiters can not be a blank(' ') character.
2. A blank(' ') can not be contained within a numeric field.
3. Fields in the FROMFILE that are longer than the TOFILE's fields will be truncated(right).
4. If the data of the FROMFILE does not represent all the fields in the TOFILE, the fields of the TOFILE will be set to null. If this happens and the TOFILE fields do not allow a null value, an error will occur and the record will not be copied to the TOFILE, unless RPLNULLVAL(*FLDDFT) is specified. *FLDDFT will allow the field default value to be inserted in place of the null value.
5. A null field in the FROMFILE can be specified by 2 adjacent field delimiters, 2 adjacent string delimiters, a field delimiter followed by a record delimiter, or a field of all blanks(' ').

Notes For Fixed Data:

The Field Definition File has the following format:

The following example is described:

Field Definition File to describe
fixed formatted file:

```
- *****/
- **** Field Definition File */
- *****/
- Description: This Field Definition */
- File defines the import's file */
- (FROMFILE) field start and */
- end positions. */
- *****/
```

```
FIELD1 1 12 13
```

```

FIELD2 14 24 0
FIELD3 25 55 56
FIELD4 78 89 90
FIELD5 100 109 0
*END

```

The following is a brief explanation of the Field Definition File format:

```

-      = Comment line
*END  = End of definition

```

Field Name	Starting Position	Ending Position	Null Char Position
FIELD1	1	12	13
FIELD2	14	24	None
FIELD3	25	55	56
FIELD4	78	89	90
FIELD5	100	109	None

The name of the field. This name is the name of the TOFILE field name.

Starting Position is the starting position for the field in the import file of each record. This is the byte position.

Ending Position is the ending position for the field in the import file of each record. This is the byte position.

Null Character Position is the position for the NULL value for the field in the import file of each record. The value 0 is defined to be that there is not a value for the NULL. The value in the import file can be 'Y' or 'N'.



'Y' means the field is NULL.
 'N' means the field is not NULL.

Each column must be separated by a blank character.

Required Parameters

FROMSTMF

Specifies the path name of the stream file from which data is copied. Either this parameter or the FROMFILE parameter is required. See path names for more information on specifying path names.

from-file-path-name: Specify the path name of the input stream file. Note: If the stream file is not in the QSYS.LIB  or independent ASP QSYS.LIB  file system, a temporary physical file will be created to contain the data of the stream file. This temporary file will be created in QRECOVERY and named QACPXXXXXX, where XXXXXX is a named generated by the system. The data will then be copied from the temporary file to the TOFILE. After the copy completes, the temporary file will be deleted.

FROMFILE

Specifies the qualified name of the from file that contains the records being copied.

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.

file-name: Specify the name of the file that contains the records to be copied.

Element 2: From File Member

Specifies the from member name.

***FIRST:** The first member (in order of creation date) of the from file is used.

Specifying *FIRST is not allowed if the FROMFILE file has no members, unless a member name was specified on an OVRDBF (Override Database File) command for the FROMFILE file.

member-name: Specify the name of the file member to receive the copied records. If a member with the specified name does not already exist in the file, the member will be created.

The from file can be any one of the file types:

- source physical file.
- DDM file.
- distributed physical file.
- program described physical file.
- single format logical file.
- physical file with 1 field that has a non-numeric field.
- tape file.

TOFILE

Specifies the name of the output database file and member that receives the copied records. This parameter is required.

Element 1: To 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.

file-name: Specify the name of the file that receives the copied records.

Element 2: To File Member

Specifies the member name of the output file to receive the copied records.

***FIRST**: The first member (in order of creation date) of the output file is used.

Specifying *FIRST is not allowed if the TOFILE has no members, unless a member name was specified on an OVRDBF (Override Database File) command for the TOFILE.

member-name: Specify the name of the file member to receive the copied records. If a member with the specified name does not already exist in the file, the member will be created.

The TOFILE can be any one of the file types:

- source physical file.
- DDM file.
- distributed physical file.
- program described physical file.
- externally described physical file.

MBROPT

Specifies whether the copy operation replaces, adds, or updates the records in a database file member if a member with the specified name already exists. If the member does not exist, it is created and added to the database file.

Note:

If *ADD or *UPDADD is specified and the TOFILE contains no records, the copy operation completes normally. If *REPLACE is specified and the TOFILE contains no records, the copy operation ends abnormally.

***ADD**: The copied records are added to the end of the existing member records.

***REPLACE**: The copied records replace the existing member records.

***UPDADD**: The system updates the duplicate key records and adds the new records to the end of the existing records. Additional information is available in the File Management topic in the Information Center.

STMFRCDLEN

The maximum record length of any record of the stream file when the DTAFMT(*DLM) is specified, or the actual record length of all the records of the stream file when DTAFMT(*FIXFLD) is specified.

***TOFILE**: The record length of the TOFILE record is used.

record-length: The length of the record of the stream file.

FROMCCSID

Specifies the FROMFILE's CCSID.

***FILE**: The CCSID of the FROMFILE's CCSID is used. If the FROMFILE is a tape file, the job's default CCSID is used.

CCSID value: The CCSID the data should be copied from if the FROMFILE's CCSID is 65535, or a tape file. If the FROMFILE's CCSID is not 65535, or the FROMFILE is not a tape file, an error condition is created.

RCDDLML

Specifies the record delimiter of the from file.

***EOR:** End of record.

***ALL:** Any single or double character combination of carriage-return and line-feed.

***CRLF:** Carriage-return followed by line-feed.

***LF:** Line-feed.

***CR:** Carriage-return.

***LFCR:** Line-feed followed by carriage-return.

***EOR:** End of record.

end-of-line-character: Specify the single character which indicates the end of a single record.

DTAFMT

Specifies the format of the data in the from file.

***DLM:** The data contains delimiter characters. Refer to parameter descriptions for STRDLM, FLDDLML, and RCDDLML for information on string, field, and record delimiter characters.

***FIXED:** The data format is fixed. The data is in fixed columns in each record. The description of the format of the data is contained in the file member identified by the FLDDFNFILE parameter.

STRDLM

Specifies the string delimiter for the data of the fields being copied from. This character indicates the start and end of character, date, time, and timestamp strings in the from file. Depending on the utility used to create the from file, some types of strings may appear in the from file without string delimiter characters.

''': A double quote (") is used as the string delimiter.

***NONE:** No delimiter is expected as the string delimiter. The blank character () represents the *NONE value.

character-value: Specify the character value for the string delimiter.

RMVBLANK

Specifies whether leading blanks are removed or retained on character fields when STRDLM(*NONE) is specified.

***LEADING:** Leading blanks are removed.

***NONE:** The leading blanks are retained on character fields when STRDLM(*NONE) is specified.

FLDDLML

Specifies the field delimiter for the record being copied from. This value is used to determine where one field ends and the next field begins.

',': The comma character is the default name of the field delimiter.

***TAB:** The tab character is the delimiter.

character-value: Specify the character value for the field delimiter.

FLDDFNFILE

Specifies the field definition file which defines the format of the data when DTAFMT(*FIXFLD) is

specified. For details on the required format for this file, refer to the File Management topic in the Information Center. If DTAFMT(*FIXFLD) is specified, this parameter is required.

Element 1: Field Definition File

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.

file-name: Specify the name of the file that contains the fixed field definition.

Element 2: Field Definition File Member

Specifies the database file member name of the field definition file.

***FIRST:** The first member (in order of creation date) in the field definition file is used.

member-name: Specify the name of the field definition file member to use. The field definition file can be any one of the file types:

- source physical file.
- DDM file.
- distributed physical file.
- program described physical file.
- externally described physical file with 1 field.

DECPNT

Specifies the decimal point character to be used when copying numeric data from the from file.

***PERIOD:** A period (.) is used for the decimal point character.

***COMMA:** A comma (,) is used for the decimal point character.

DATFMT

Specifies the date format to be used when copying date fields from the from file.

***ISO:** The International Organization for Standardization (ISO) date format **yyyy-mm-dd** is used.

***USA:** The United States date format **mm/dd/yyyy** is used.

***EUR:** The European date format **dd.mm.yyyy** is used.

***JIS:** The Japanese Industrial Standard date format **yyyy-mm-dd** is used.

***MDY:** The date format **mm/dd/yy** is used.

***DMY:** The date format **dd/mm/yy** is used.

***YMD:** The date format **yy/mm/dd** is used.

***JUL:** The Julian date format **yy/ddd** is used.

***YYMD:** The date format **yyyymmdd** is used.

DATSEP

Specifies the date separator for the date format. The separator is ignored for DATFMT of *ISO, *USA, *EUR, and *JIS because these formats have a fixed date separator.

/: A forward slash (/) is used as the date separator character.

-: A dash (-) is used as the date separator character.

.: A period (.) is used as the date separator character.

,: A comma (,) is used as the date separator character.

***BLANK**: A blank () is used as the date separator character.

TIMFMT

Specifies the time format to be used when copying time fields from the from file.

***ISO**: The International Organization for Standardization (ISO) time format **hh.mm.ss** is used.

***USA**: The United States time format **hh:mmxx** is used, where **xx** is AM or PM.

***EUR**: The European time format **hh.mm.ss** is used.

***JIS**: The Japanese Industrial Standard time format **hh:mm:ss** is used.

***HMS**: The **hh:mm:ss** format is used.

TIMSEP

Specifies the time separator for the time format. The parameter is ignored if TIMFMT is *ISO, *USA, *EUR, or *JIS because these formats define the required time separator character.

:: A colon (:) is used as the time separator character.

.: A period (.) is used as the time separator character.

***BLANK**: A blank () is used as the time separator character.

FROMRCD

Specifies which records are copied from the from file.

Element 1: Starting Record Number

***START**: The copy operation begins with the first record in the from file.

starting-record-number: Specify a record number, ranging from 1 through 4294967288, that identifies the first record copied from the from file. If both FROMRCD and TORCD record number values are specified, the FROMRCD value must be less than or equal to the TORCD value.

Element 2: Number of Records to Copy

Specifies the number of records to be copied from the from file.

***END**: Records are copied until the end-of-file condition is indicated.

number-of-records: Specify a number, ranging from 1 through 4294967288, that identifies the number of records to be copied from the from file. If an end-of-file condition is reached before this number of records have been copied, no error messages are issued.

ERRLVL

Specifies the maximum number of recoverable read or write errors for the TOFILE that are tolerated during the copy operation.

***NOMAX**: No maximum number of errors is specified, and all recoverable errors are tolerated. The copy operation continues regardless of the number of recoverable errors found.

0: No errors are allowed.

number-of-errors: Specify a value that specifies the maximum number of recoverable errors allowed. If one more error occurs than the value specified here, the copy operation

ERRRCDFILE

Specifies the database file where the records that are in error should be written.

***NONE:** No error record file is provided.

Element 1: Error Record 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.

file-name: Specify the error record file name.

Element 2: Error Record File Member

Specifies which member of the error file is used to contain the from file records which contained errors.

***FIRST:** The first member (in order of creation date) in the error file is used.

member-name: Specify the member name of the file. The error record file can be any one of the file types:

- source physical file.
- DDM file.
- distributed physical file.
- program described physical file.
- externally described physical file.

ERRRCDOPT

Specifies how error records are added to the error record file.

***ADD:** The system adds the new records to the end of the existing records.

***REPLACE:** The system deletes any existing records and adds the new records.

RPLNULLVAL

Specifies whether null field values will be replaced when copying import file records.

***NO:** Null values will not be replaced. If a null value is detected when parsing an import file record, an error message is sent and the copy operation fails.

***FLDDFT:** If a null value is detected when parsing an import file record, the corresponding field in the database file record is assigned a default value based on the field type or DDS default value.

» IDCOL

Specifies if the to-file is an SQL table which contains a column with the IDENTITY attribute or a column with the ROWID data type, whether the value for the column will be generated by the system or the default value is used.

***GEN:** A system-generated value will be inserted into the Identity Column or ROWID column.

***FROMFLD:** If a value exists in the Identity Column or ROWID column of the fromfile field, this value will be inserted into the Identity Column of the to-file. <<

Examples for CPYFRMIMPF

Example 1: Copying Physical File Import File

```
CHGQRYA DEGREE(*NBRTASKS 3)

CPYFRMIMPF FROMFILE(IMPFILE) TOFILE(DB2FILE)
  FLDDLML(';') RCDDLML(X'07')
  DATFMT(*JIS) TIMFMT(*JIS)
```

The Change Query Attribute (CHGQRYA) is run prior to CPYFRMIMPF to allow the copy processing to be done by three tasks running in parallel.

All records of from file IMPFILE will be copied to the externally described physical file DB2FILE. Fields in the from file are delimited by semi-colon (;) characters. Each record in the from file is delimited by a hexadecimal '07' character. Input date fields are in **yyyy-mm-dd** format. Input time fields are in **hh:mm:ss** format.

Example 2: Copying Tape File Import File

```
OVRTAPF FILE(QTAPE) DEV(TAP02) SEQNBR(3)

CPYFRMIMPF FROMFILE(QTAPE) TOFILE(DB2WHS)
  ERRFILE(IMPERR)
```

The Override Tape File (OVRTAPF) parameter is run prior to CPYFRMIMPF to indicate that tape device TAP02 should be used for doing the copy. The from file must be the third file on the tape mounted on TAP02.

All records of the from file will be copied to the externally described physical file DB2WHS. Fields in the from file are delimited by comma (,) characters. Input date fields are in **yyyy-mm-dd** (ISO) format. Input time fields are in **hh.mm.ss** (ISO) format. From file records that are found to contain errors and cannot be added to file DB2WHS are added to error file IMPERR.

Error messages for CPYFRMIMPF

*ESCAPE Messages

CPF2817

Copy command ended because of error.



Printed in U.S.A.