# Predict Application Control

## PAC Reference

Version 2.6.1

November 2016

This document applies to Predict Application Control Version 2.6.1.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.



**Document ID: PAC-REF-261-20161120**

## Table of Contents

# PAC Reference

This documentation is intended for all users of Predict Application Control.

It is organized under the following headings:

| | |
|---|---|
| **Command Reference** | This document describes the available direct commands which can be issued from any screen in the PAC system to bypass menus and access the particular screens directly. |
| **Migration Utility** | This document describes the PAC Migration utility which may be executed during migration events, using the command MIGLOAD for loading Natural objects into PAC. |
| **Compare Utility** | This document contains information on the PAC Compare utility. The comparing of Natural, Predict and foreign objects both under the control of PAC, as well as outside of the PAC environment is described. |
| **Scan Utilities** | This document describes both the scan utility for migration path verification and default update and the scan utility for Natural object versions. |
| **User Exits** | This document provides detailed information on PAC user exits, whereby control can be passed to user-written Natural programs (user exit routines) to override the change management procedure defaults established by PAC. |
| **System Applymods** | This document describes the applymods currently used by PAC. |
| **Object Type Codes** | This document lists and describes the object type codes used in PAC. |
| **Application Programming Interface** | This document describes PAC's Application Programming Interface (API) facilities, which are Natural subprograms invoked using CALLNAT statements to access PAC directly from within a user-written program without invoking PAC with MENU, ADMIN, or SYSPAC commands. |

# I

# 1 Command Reference

Instead of using function codes, you can issue direct commands from any screen in the PAC system to bypass menus and access the particular screens directly.

Direct commands are either Natural (N) commands or commands that are specific to PAC (P). The ADMIN and MENU commands are both Natural and PAC commands.

This chapter covers the following topics:

## SYSPAC Commands - N

The direct commands in this section are used to execute SYSPAC functions. These commands may be issued either when SYSPAC is invoked or from a menu screen.

Direct commands are issued in either interactive or batch mode.

### To issue commands in interactive mode

- from any library, enter SYSPAC followed by a direct command string.

  After the direct command is issued, you are usually returned to the library from which you issued the command.

  Or:

  from library SYSPAC, enter MENU followed by a direct command string.

  Or:

  from a SYSPAC menu screen, enter a direct command string.

### To issue commands in batch mode

- from any library, enter SYSPAC followed by a direct command string.

  Or:

  from library SYSPAC, enter MENU followed by a direct command string.

**Notes:**

1. All direct commands entered in batch must be entered in delimiter mode when a startup of MENU is specified for the applications SYSPAC and SYSPAA.
2. The commands **ADD**, **LINK**, **PURGE** and **RENAME** can only be issued for entity-type EVENT in batch.

This section covers the following topics:



## Command Syntax

The generalized syntax for SYSPAC direct commands is as follows:

```
COMMAND ENTITY-TYPE1 entity-name1 [[operator][ ENTITY-TYPE2 ] entity-name2 ]
```

The command and entity-type arguments are always entered in capital letters. A space or a delimiter (the default is a comma) separates the arguments. The delimiter must be used for batch commands.

The arguments enclosed in square brackets in the syntax statement are optional for most commands, but when used, their values are restricted by the first set of arguments.

The characters underlined in each command's syntax statement may be used alone to represent the command and entity-type.

### Operators

The operator argument is optional, but if the operator is specified, the correct operator for the selected command must be used.

### Entity Types and Entity Names

Only certain entity-types are valid for a particular command. Valid values for each argument are discussed for each command in the following sections. The entity-name must correspond to the associated entity-type designation.

The possible entity-types for PAC commands are as follows:

- APPLICATION
- EVENT
- JOB
- KEYWORD
- OBJECT
- REQUEST
- STATUS

**Note:** Direct commands are not currently available for the entity-type keyword.

The format for the entity-names of each of the entity-types is A32 meaning the name is alphanumeric and can consist of up to 32 characters.

## Command Processing

Most direct commands can be issued from the command line of any PAC screen by entering the desired command in accordance with the syntax described in this section.

The command bypasses the PAC menus and takes you directly to the desired function screen. For example, if you wish to display an application status link from the Migration Event Menu, enter the following direct command on the command line:

```
DISPLAY STATUS development FOR APPLICATION ord-exam
```

If the entity name is not found, the maintenance menu of the specified entity type (application, event, job, keyword, request, status) is displayed with a message at the bottom of the screen stating that the entity name specified was not found. From this screen, you may request a list of valid entity names by entering one of the following in the relevant entity type field:

- An asterisk (*);

- A range of valid entity names;
- The first one or two characters of an entity name followed immediately by the asterisk notation (for example, "ord*").

Refer to the PAC User's Guide, section Using Range Notation in section Navigating in PAC for more information.

## ADD Command

ADD *ENTITY-TYPE1 entity-name1* [TO][ *ENTITY-TYPE2* ] *entity-name2* ]

Valid entity-types for the ADD command include the following:

| **entity-type1** | **entity-type2** |
|---|---|
| APPLICATION | STATUS |
| EVENT | - |
| JOB | - |
| KEYWORD | - |
| REQUEST | - |
| STATUS | APPLICATION |

The following syntax is used to add (link) an application to a status:

```
ADD APPLICATION application-name [TO][STATUS]status-name
```

The following syntax is used to add (link) a status to an application:

```
ADD STATUS status-name [TO][APPLICATION]application-name
```

The remaining valid values for entity-type1 are not used with a TO-clause.

### ADD Command Examples

```
ADD APPLICATION ord-exam
ADD APPLICATION ord-exam TO STATUS user-test
ADD APPLICATION ord-exam STATUS user-test
ADD APPLICATION ord-exam user-test
ADD EVENT ord-dc-01
ADD JOB os-export-jcl
ADD KEYWORD project1
ADD REQUEST fix-01
ADD STATUS user-test
ADD STATUS user-test TO APPLICATION ord-exam
ADD STATUS user-test APPLICATION ord-exam
ADD STATUS user-test ord-exam
```

### ADD EVENT Batch Command

The command syntax for adding a migration event in batch is as follows:

```
ADD,EVENT,event-name,[FOR] application-name,[FROM] origin-status,
  [TO] destination-status
  [,SCHEDULED,yy-mm-dd,[AT] hh:ii:ss]
  [,MREQ,maintenance-request]
  [,ARCEVENT,archive-event-name]
  [,GENTYPE,C/I/M/P/R/S/U/L]
  [,SET identifiers required for GENTYPE C or S]
  [,WITH clause optional for GENTYPE P]
```

The values for the optional keywords are described in the following table:

| Keyword | Value |
|---|---|
| FOR | name of the application |
| FROM | name of the origin status |
| TO | name of the destination status |
| SCHEDULED | date (yy-mm-dd format) and time (hh:ii:ss format) that the application is to become active in production. |
| REPLACE | replace existing object(s) with the version being migrated. |
| MREQ | name of the maintenance request associated with the migration event (if applicable). |
| ARCEVENT | used if you are restoring archived objects to PAC. ARCEVENT is the name of the archive event that initially archived the objects. |
| GENTYPE | one-character code (C, I, M, P, R, S, U, L) for the type of object list to be generated. Refer to the following GENTYPE Option sections for more information.<br>Note: You must specify a GENTYPE option to generate the object list. Otherwise, you must create the object list manually. |

#### GENTYPE C: Predict Case Generate

The object list is generated from a Predict Case set. The SETUID (user ID of the set; up to eight characters) and SETNAME (name of the set; up to 20 characters) are required when GENTYPE C is specified.

,SETUID, *pcaset-userID*

,SETNAME, *pcaset-name*

The set identified must be defined on the application status link with an origin status of maintenance or development; it must be valid for the current NTFILE definition for Predict Case in the Natural parameter modules.

### GENTYPE Option I: Input

The object list or list of URLs (denoting the objects to be transferred into PAC) is read from instream input with the GENTYPE I option. The following example shows how this facility may be used.

Work file 1 (CMWKF01) is not used; the data read for creating the object list is contained in the primary input file (CMSYNIN).

The format of the input data must be as follows:

*OBJECT-NAME* (A32), *OBJECT-TYPE* (A4) [ ,*VERSION-NUMBER* (N4) ,*STATUS* (A32) ]

The data fields must be separated with a comma (,). The end of the list must be designated with a period (.). If the version number is omitted, the most recent version of the object will be included on the object list.

For example, CMSYNIN contains the following commands and data:

```
LOGON SYSPAC
MENU
ADD,EVENT,MAY10-01,FOR,ORDERS,FROM,CONTROL,TO,PRODUCTION,%
GENTYPE,I
ORD-COPY,C,2
ORD-CUST,N,1
ORD-GLOB,G,3
ORD-MAP,M,3
ORD-MAIN,P,1
.
FIN
```

The percent sign (%) indicates the end of input for the current line and that the remainder of the command is continued on the next line.

The above example creates a migration event MAY10-01 for the application ORDERS from the status CONTROL to the status PRODUCTION with the resulting object list:

```
ORD-COPY,C,0002
ORD-CUST,N,0001
ORD-END,N,0003
ORD-GLOB,G,0003
ORD-MAP,M,0003
ORD-MAIN,P,0001
```

### ADD EVENT Batch Command Example

```
//@USER JOB @ACCOUNT,CLASS=A
```

```
//EVENT EXEC PACBATCH
//CMSYNIN DD *
LOGON SYSPAC
MENU
ADD,EVENT,new-event@GEN,FOR,application,%
FROM,development,TO,user-test,%
GENTYPE,I
program1,P
program2,P
program3,P
program4,P
program5,P
```

#### GENTYPE Option M: Maintenance Request Generate

The object list will be built from a maintenance request. If MREQ is not specified, the object list will be built from the maintenance request associated with the event name. The object list is built from objects previously checked-out of PAC to a maintenance status. PAC selects checked-out objects with the same maintenance request ID, application, and check-out library (of the From/origin status path) as the event.

For a maintenance request with multiple applications: When adding an event in batch, or generating a list from a maintenance request that was used to check out objects for more than one application, you may add one event for each application in use by the maintenance request using the "@GEN" string for the event name and an asterisk (*) for the application name. If a fixed name is used for the application, then an event for only that application is created. If the "@GEN" string is not used in the event name, then an event is created for the first application only.

**Note:** This is valid only if origin status is neither of type Incorporation nor of type Archive.

#### GENTYPE Option P: PAC Generate

The object list will be built based on the objects currently in the origin status compared with objects in the destination status of the various types of events:

- Development, Maintenance or Incorporation
  An object is considered in the list only if the object does not exit for the same application. In the case of Development or Maintenance, objects with sources in the original location are taken into consideration. In the case of Incorporation, objects with sources and/or loadables in the original location are taken into consideration.
- Restoration
  Object list is generated only if an archiving event in the Archive Event field is also specified.
- Control
  A versioned object will be placed in the list if the object has not already been migrated, overwritten or retired. This is valid for all destination statuses except Archive.
- Archive

If the event is one of archiving, the Natural objects in the compartment whose archiving with removal from the compartment (archiving with finalization) is allowed by the application's Retention Parameters (described in subsection Retention Parameters) are put on the list.

- Retirement
  Objects which were never overwritten or retired from a test, maintenance or production environment will be placed on the list.
- De-Archiving
  Object list is generated if an archiving event in the Archive Event field is also specified.
- Retirement from CONTROL
  If the event is one of retirement from CONTROL, the objects whose retirements are allowed by the application's Retention Parameters (described in subsection Retention Parameters) are put on the list.
- Alignment
  Highest numbered Natural object versions are placed on the list.
- False Migration
  Objects migrated to the environment defined in the application (origin status link) and have not been overwritten nor retired are placed on the list.

For an origin status of an Archive status type, the Archive event used to archive the objects should be specified.

When you specify P, you may optionally add the following WITH clause to specify selection criteria for the object list:

```
WITH  [ ,NAME, object-name
        ,FMDATE, yy-mm-dd
        ,FMTIME, hh:ii:ss
        ,TODATE, yy-mm-dd
        TOTIME, hh:ii:ss
        TYPE, A | L | M | N | P | S... ]
```

NAME includes only the objects in the specified range.

USER includes only the objects saved by a user in the NAME range.

TYPE is a single character designation for any valid Natural object type except error message.

Dates and times must be entered in the formats given. Only objects compiled since the FM (from) date and time and before the TO date and time will be included in the list.

For dates

- yy is the last two digits of the year (e.g., 97 for 1997), mm is the month (e.g., 02 for February), and dd is the day of the month.

- If left blank, the year default is 00; the month default is 01; and the day default is 01.

  For times

- hh is the hour of the day using a 24-hour clock; ii is the minutes of an hour (from 01-60); and ss is the seconds of a minute (01-60).
- If left blank, the hour default is 00; the minute default is 01; and the second default is 01.

#### GENTYPE Option R: Work File 1

You may use the ADD EVENT command in batch to read object lists or list of URLs (denoting the objects to be transferred into PAC) from external input. For example, objects placed in a work file (Work File 1) may become the object list.

The maximum length for the URL-NAME is 250 and must start with `http://` or `https://`.

The following example shows how this facility may be used.

Work File 1 (label CMWKF01) is required and contains the data from which the object list will be generated.

The format of the data of Work File 1 (CMWKF01) must be as follows:

*OBJECT-NAME* (A32), *OBJECT-TYPE* (A4) [ ,*VERSION-NUMBER* (N4) ,*STATUS* (A32) ]

The data fields must be separated with the specified input delimiter, in this case, the comma (,). If the version number is omitted, the most recent version of the object will be included in the object list. For example:

```
ORD-COPY,C,2
ORD-CUST,N,1
ORD-END,N,3
ORD-GLOB,G,3
ORD-MAP,M,3
ORD-MAIN,P,1
```

The primary input file (label CMSYNIN) contains the following commands:

```
LOGON SYSPAC
MENU
ADD,EVENT,MAY10-01,FOR,ORDERS,FROM,CONTROL,TO,PRODUCTION,%
GENTYPE,R
```

The percent sign (%) tells the user to put the input command to PAC over two lines.

The above example creates a migration event MAY10-01 for the application ORDERS to be migrated from the status CONTROL to the status PRODUCTION with the resulting object list:

```
ORD-COPY,C,0002
ORD-CUST,N,0001
ORD-END,N,0003
ORD-GLOB,G,0003
ORD-MAP,M,0003
ORD-MAIN,P,0001
```

#### GENTYPE Option S: Predict Set Generate

With GENTYPE S, the object list is built from a Predict set identified by user ID (SETUID), set number (SETNUM), and file number (SETLOC).

```
,SETUID, prdset-userID
,SETNUM, prdset-number
,SETLOC,REMOTE/LOCAL
```

SETLOC is the location of Predict cross-reference data. For objects from the Control status, the location is always local (that is, the set is stored within PAC). Otherwise, the set location is always remote (that is, the set is stored in the Predict file specified by the application status link).

#### GENTYPE Option U: PAC Generate (from origin status)

With GENTYPE U, the object list is built from a URL specifying a source in a repository of an external version control software. This option is used similarily to the SEL U command with the exception that you can not expand a specific node residing below a certain root URL any further. Instead, all content below the specified URL is taken as input for the object list. However, specifying a certain pattern (file name and or extension), a certain date or time or a period of time is also possible.

The following batch parameters can be used:

| Parameter | Explanation |
|---|---|
| URL-PART-1 | The first part of the URL. The number of characters allowed for a URL is 250. However, since the number of characters per parameter is limited to 32 plus a delimiter character (a % sign), you can specify up to eight URL-PART parameters (URL-PART-1 to URL-PART-8) with each parameter containing a part of the complete URL and the delimiter character. |
| FILE-PATTERN | This parameter can contain a file name or part of the file name with an asterisk as wildcard character. |

| Parameter | Explanation |
|---|---|
| EXT-PATTERN | This parameter can contain a file extension or part of the file extension with an asterisk as wildcard character. |

**Example**

```
PAC V26,DNAT1
MENU
ADD,EVENT,HEA-&GEN,FOR,NAT_TEST,FROM,NAT_EXT %
TO,CONTROL %
GENTYPE,U %
NAME,SVN-L* %
URL-PART-1, http://natcvs/repos/pac/ %
URL-PART-2, test/NAT-FUSER/NAT-APIT/SRC/ %
FILE-PATTERN, ABC*
EXT-PATTERN, NSN
FIN
```

#### GENTYPE Option L: Locks

This option generates a migration list consisting of one entry for each locked entity, or, for short, 'lock', of the specified type and pertaining to the relevant application. There are five lock types to choose from:

| Type | Explanation |
|---|---|
| O | object records |
| N | new object - status links |
| H | historical object - status links corresponding to objects just superseded |
| R | historical object - status links corresponding to objects just retired from non-ACF locations |
| S | all object - status links |

The syntax for the direct command is as follows:

```
<OBJECT NAME,<OBJECT TYPE>
or
<OBJECT NAME>,<OBJECT TYPE>,<VERSION NUMBER>
```

depending on whether the generation is "without version" or with, respectively.

### AUTHORIZE EVENT Batch Command

**Note:** The migration event must be in a Pending or Validated state. The authorization request will take a Pending event through the Validated state automatically.

The AUTHORIZE EVENT direct command may be used to authorize migration events in batch.

The syntax for the direct command is as follows:

```
AUTHORIZE, EVENT, event-name
    [ ,JOB, job-name
      ,EXPAND,{C | E | R | U | N}
      ,{COPY | MOVE | INCLUDE}
      ,REPLACE,{Y | N}
      ,APPLYMODS, applymod-number(s)
      ,FTT, ftt-name
      ,EXPAND-STATUS, status-name
      ,WORKFILE,{Y | N}
      ,{BATCH | ONLINE}
      ,{CAT,{D | Y | N}} ]
```

AUTHORIZE EVENT Batch Command Example:

```
LOGON SYSPAC
MENU
AUTHORIZE,EVENT,ord dc 02,JOB,natural migrate workfile,EXPAND,C
AUTHORIZE,EVENT,ord dc 03,MOVE,REPLACE,Y,APPLYMODS,4,5
```

### BACKOUT EVENT Batch Command

The BACKOUT EVENT direct command may be used to back out a migration event in batch.

The syntax for the direct command is as follows:

```
BACKOUT,EVENT, event-name
```

BACKOUT EVENT Batch Command Example:

```
//@USER   JOB @ACCOUNT,CLASS=A,MSGLCASSS=X,REGION=2M
//BACKOUT EXEC PACBATCH
//CMSYNIN DD *
LOGON SYSPAC
MENU
BACKOUT,EVENT,ord_dc_02
FIN
/*
```

### COPY Command

```
COPY ENTITY-TYPE1 entity-name1a [TO] entity-name1b
```

Valid entity-types for the COPY command include the following:

**entity-type 1**

EVENT

JOB

KEYWORD

REQUEST

The COPY command requires a TO-clause, which consists of just an entity-name, with or without the TO operator. The corresponding entity type is always entity-type1.

Examples:

```
COPY EVENT ord-dc-01 TO ord-dc-02
COPY JOB os-export-jcl new-export-jcl
COPY KEYWORD project1 TO project2
COPY REQUEST fix-01 fix-02
```

### DISPLAY Command

**Note:** Refer to the following section for information about the DISPLAY OBJECT command.

```
DISPLAY ENTITY-TYPE1 entity-name1 [[FOR][ ENTITY-TYPE2 ] entity-name2 ]
```

Valid entity-types for the DISPLAY command include the following:

| entity-type1 | entity-type2 |
|---|---|
| APPLICATION | STATUS |
| EVENT | - |
| JOB | - |
| KEYWORD | - |
| REQUEST | - |
| STATUS | APPLICATION |

The following syntax is used to display an application linked to a status:

```
DISPLAY APPLICATION application-name [FOR][STATUS]status-name
```

The following syntax is used to display a status linked to an application:

```
DISPLAY STATUS status-name [FOR ][APPLICATION]application-name
```

The remaining valid values for entity-type1 are not used with a FOR-clause.

Examples:

```
DISPLAY APPLICATION ord-exam
DISPLAY APPLICATION ord-exam FOR STATUS user-test
DISPLAY APPLICATION ord-exam STATUS user-test
DISPLAY APPLICATION ord-exam user-test
```

```
DISPLAY EVENT ord-dc-01
DISPLAY JOB os-export-jcl
DISPLAY KEYWORD project1
DISPLAY REQUEST fix-01
```

```
DISPLAY STATUS user-test
DISPLAY STATUS user-test FOR APPLICATION ord-exam
DISPLAY STATUS user-test APPLICATION ord-exam
DISPLAY STATUS user-test ord-exam
```

### DISPLAY OBJECT Command

```
DISPLAY OBJECT object-name [ version | status ] [[FOR][APPLICATION] application-name
```

The only valid entity-types for this particular DISPLAY command are as follows:

**entity-type1entity-type2**

OBJECT APPLICATION

Note that a specific version of the object may be specified for display. You may also, or alternatively, specify the status for which you want the object version(s) displayed.

Examples:

```
DISPLAY OBJECT ord-exam
DISPLAY OBJECT ord-main 0001
DISPLAY OBJECT ord-main user-test
DISPLAY OBJECT ord-main 0001 user-test
```

```
DISPLAY OBJECT ord-main FOR APPLICATION ord-exam
DISPLAY OBJECT ord-main APPLICATION ord-exam
DISPLAY OBJECT ord-main ord-exam
```

```
DISPLAY OBJECT ord-main 0001 FOR APPLICATION ord-exam
DISPLAY OBJECT ord-main 0001 APPLICATION ord-exam
DISPLAY OBJECT ord-main 0001 ord-exam
```

```
DISPLAY OBJECT ord-main user-test FOR APPLICATION ord-exam
DISPLAY OBJECT ord-main user-test APPLICATION ord-exam
DISPLAY OBJECT ord-main user-test ord-exam
```

### DISPLAY OBJECT LAST Command

You may display the most current version of an object, or the next to most current version of a specified object:

```
DISPLAY OBJECT obj-name LAST - n [ version | status ] [[FOR][APPLICATION] application-name
```

where - *n* is the most current version minus the number of versions.

Examples:

```
DISPLAY OBJECT ord-main LAST APPLICATION ord-exam
DISPLAY OBJECT ord-main LAST - 1 APPLICATION ord-exam
```

### GENLIST EVENT Batch Command

You may use the GENLIST EVENT batch command to create the object list for an existing migration event and to import lists of URLs (denoting the objects to be transferred into PAC) either instream (option I of GENTYPE) or by means of a workfile (option R of GENTYPE).

The GENTYPE options available are the same as those for the **ADD EVENT** command.

If an object list already exists for the event, it will be replaced.

The syntax for the GENLIST EVENT command is as follows:

```
GENLIST,EVENT,event-nameGENTYPE,C/I/M/P/R/S/U/W/L
  [,MREQ,maintenance-request]
  [,ARCEVENT,archive-event-name]
  [,SET identifiers required for GENTYPE C or S]
  [,WITH clause optional for GENTYPE P]
```

The values for the optional keywords are described in the following table:

| Keyword | Description |
|---|---|
| GENTYPE | one-character code (C, I, M, P, R, S, U, W, L) for the type of object list to be generated. Refer to the GENTYPE Option sections associated with the ADD EVENT command. In addition, the GENLIST command provides the GENTYPE W (Write to Output File) option. Use this code to write the object list of an existing event to an external output file (Work File 1 is required and must be a minimum length of 72 bytes per record).<br>Note: You must specify a GENTYPE option to generate the object list. |
| MREQ | name of the maintenance request associated with the migration event (if applicable). |
| ARCEVENT | used if you are restoring archived objects to PAC. ARCEVENT is the name of the archive event that initially archived the objects. |

Example:

```
LOGON SYSPAC
MENU
GENLIST,EVENT,batch-add-d-ut,GENTYPE,R
```

## LINK Command

```
LINK ENTITY-TYPE1 entity-name1 [TO] [ ENTITY-TYPE2 ] entity-name2
```

The LINK command is used to link applications and statuses. Valid entity-types for the LINK command include the following:

| **entity-type1** | **entity-type2** |
|---|---|
| APPLICATION | STATUS |
| STATUS | APPLICATION |

At least the entity-name argument of the TO-clause is required.

The following syntax is used to link an application to a status:

```
LINK APPLICATION application-name [TO][STATUS]status-name
```

The following syntax is used to link a status to an application:

```
LINK STATUS status-name [TO][APPLICATION]application-name
```

Examples:

```
LINK APPLICATION ord-exam TO STATUS user-test
LINK APPLICATION ord-exam STATUS user-test
LINK APPLICATION ord-exam user-test
```

```
LINK STATUS user-test TO APPLICATION ord-exam
LINK STATUS user-test APPLICATION ord-exam
LINK STATUS user-test ord-exam
```

## MODIFY Command

MODIFY *ENTITY-TYPE1 entity-name1* [[FOR] [ *ENTITY-TYPE2* ] *entity-name2* ]

Valid entity-types for the MODIFY command include the following:

| **entity-type1** | **entity-type2** |
|---|---|
| APPLICATION | STATUS |
| EVENT | - |
| JOB | - |
| KEYWORD | - |
| REQUEST | - |
| STATUS | APPLICATION |

The following syntax is used to modify an application linked to a status:

```
MODIFY APPLICATION application-name [FOR ][STATUS]status-name
```

The following syntax is used to modify a status linked to an application:

```
MODIFY STATUS status-name [FOR ][APPLICATION]application-name
```

The remaining valid values for entity-type1 are not used with a FOR-clause.

Examples:

```
MODIFY APPLICATION ord-exam
MODIFY APPLICATION ord-exam FOR STATUS user-test
MODIFY APPLICATION ord-exam STATUS user-test
MODIFY APPLICATION ord-exam user-test
```

```
MODIFY EVENT ord-dc-01
MODIFY JOB os-export-jcl
MODIFY KEYWORD project1
MODIFY REQUEST fix-01
```

```
MODIFY STATUS user-test
MODIFY STATUS user-test FOR APPLICATION ord-exam
MODIFY STATUS user-test APPLICATION ord-exam
MODIFY STATUS user-test ord-exam
```

## PURGE Command

PURGE *ENTITY-TYPE1 entity-name1* [[FOR] [ *ENTITY-TYPE2* ] *entity-name2* ]

Valid entity-types for the PURGE command include the following:

| **entity-type1** | **entity-type2** |
|---|---|
| APPLICATION | STATUS |
| EVENT | - |
| JOB | - |
| KEYWORD | - |
| REQUEST | - |
| STATUS | APPLICATION |

The following syntax is used to purge an application linked to a status:

```
PURGE APPLICATION application-name [FOR ][STATUS]status-name
```

The following syntax is used to purge a status linked to an application:

```
PURGE STATUS status-name [FOR ][APPLICATION]application-name
```

The remaining valid values for entity-type1 are not used with a FOR-clause.

Examples:

```
PURGE APPLICATION ord-exam
PURGE APPLICATION ord-exam FOR STATUS user-test
PURGE APPLICATION ord-exam STATUS user-test
PURGE APPLICATION ord-exam user-test
```

```
PURGE EVENT ord-dc-01
PURGE JOB os-export-jcl
PURGE KEYWORD project1
PURGE REQUEST fix-01
```

```
PURGE STATUS user-test
PURGE STATUS user-test FOR APPLICATION ord-exam
PURGE STATUS user-test APPLICATION ord-exam
PURGE STATUS user-test ord-exam
```

## RELEASE EVENT Command

The RELEASE EVENT direct command may be used to back out a migration event in batch.

This command performs the same functionality as the BACKOUT command. It is intended to remove the BACKOUT command in a future release of the product.

The syntax for the direct command is as follows:

RELEASE,EVENT, *event-name*

RELEASE EVENT Batch Command Example:

```
//@USER   JOB @ACCOUNT,CLASS=A,MSGLCASSS=X,REGION=2M
//RELEASE EXEC PACBATCH
//CMSYNIN DD *
LOGON SYSPAC
MENU
RELEASE,EVENT,ord_dc_02
FIN
/*
```

### RENAME Command

```
RENAME ENTITY-TYPE1 entity-name1a [TO] entity-name1b
```

Valid entity-types for the RENAME command include the following:

Valid entity-types for the COPY command include the following:

**entity-type1**

JOB

KEYWORD

The RENAME command requires a TO-clause, which consists of just an entity-name, with or without the TO operator. The corresponding entity type is always entity-type1.

Examples:

```
RENAME JOB os-export-jcl new-export-jcl
RENAME KEYWORD project1 TO project2
```

### RESET EVENT Command

The RESET EVENT direct command may be used to back out a migration event in batch.

This command does what the online RESET command does. When a migration event is reset, the following occurs:

- The step number of the event is reset to 1.
- The audit report is deleted.

The syntax for the direct command is as follows:

```
RESET,EVENT, event-name
```

RESET EVENT Batch Command Example:

```
//@USER   JOB @ACCOUNT,CLASS=A,MSGLCASSS=X,REGION=2M
//RESET EXEC PACBATCH
//CMSYNIN DD *
LOGON SYSPAC
MENU
RESET,EVENT,ord_dc_02
```

```
FIN
/*
```

### SELECT Command

SELECT *ENTITY-TYPE1 entity-name1* [[FOR] [ *ENTITY-TYPE2* ] *entity-name2* ]

Valid entity-types for the SELECT command include the following:

| **entity-type1** | **entity-type2** |
|---|---|
| APPLICATION | STATUS |
| EVENT | - |
| JOB | - |
| KEYWORD | - |
| REQUEST | - |
| STATUS | APPLICATION |

The following syntax is used to select an application linked to a status:

```
SELECT APPLICATION application-name [FOR ][STATUS ]status-name
```

The following syntax is used to select a status linked to an application:

```
SELECT STATUS status-name [FOR ][APPLICATION ]application-name
```

The remaining valid values for entity-type1 are not used with a FOR-clause.

Examples:

```
SELECT APPLICATION ord-exam
SELECT APPLICATION ord-exam FOR STATUS user-test
SELECT APPLICATION ord-exam STATUS user-test
SELECT APPLICATION ord-exam user-test
```

```
SELECT EVENT ord-dc-01
SELECT JOB os-export-jcl
SELECT KEYWORD project1
SELECT REQUEST fix-01
```

```
SELECT STATUS user-test
SELECT STATUS user-test FOR APPLICATION ord-exam
SELECT STATUS user-test APPLICATION ord-exam
SELECT STATUS user-test ord-exam
```

### SUBMIT EVENT Batch Command

The SUBMIT EVENT direct command may be used to submit a migration event in batch only in the z/OS environment. If you are using Natural 2.1, you must copy the module NATRJET from the Natural 2.1 distribution load library into a library in your current steplib concatenation as module NATRJE.

The syntax for the SUBMIT EVENT command is as follows:

SUBMIT, EVENT, *event-name*

Note the following when submitting a migration in batch using direct commands:

- You may submit a migration event in batch using direct commands only in the z/OS environment.
- The migration event must be in an Authorized or Started state.
- If you are using Natural 2.1, you must copy the module NATRJET from the Natural 2.1 distribution load library into a library in your current steplib concatenation as module NATRJE.
- A group event may be resubmitted in order to submit the next member event.

Example

```
//@USER   JOB @ACCOUNT,CLASS=A,MSGLCASSS=X,REGION=2M
//SUBMIT  EXEC PACBATCH
//CMSYNIN DD *
LOGON SYSPAC
MENU
SUBMIT,EVENT,ord_dc_02
FIN
/*
```

### TRY EVENT Command

The TRY EVENT direct command lists the objects whose retirement or the finalisation of whose archiving would be refused if the retirement were to be submitted or the archiving finalised at the time when the command is issued.

There can be five reasons for such refusals: four application defaults' settings and the setting (OFF) of applymod 23.

The objects get messages, but the retirement or finalisation of archiving goes on. This provides a way of knowing before the submission of the retirement or archiving event whether he was likely to run into any such refusal.

```
TRY,EVENT, event-name
```

### UNDO EVENT Command

The UNDO EVENT direct command may be used to back out a migration event in batch.

This command is applicable to emigrations to test or production and to retirements from test or production, provided the events have reached stage 6 and have not been unlocked.

The syntax for the direct command is as follows:

UNDO,EVENT, *event-name*

UNDO EVENT Batch Command Example:

```
//@USER   JOB @ACCOUNT,CLASS=A,MSGLCASSS=X,REGION=2M
//UNDO  EXEC PACBATCH
//CMSYNIN DD *
LOGON SYSPAC
MENU
UNDO ,EVENT,ord_dc_02
FIN
/*
```

## Other SYSPAC Commands

This section covers the following topics:

- ADMIN Command - N and P
- DEACTIVATE Command - P
- MENU Command - N and P
- PACADJST Command - N
- REFRESH Command - P
- SYSPAC Command - N
- SYSPACA Command - N
- UNLOCK Command - P
- Profile Command - N

**Note:** In the following paragraphs, N represents a Natural command; P represents a PAC command. ADMIN and MENU are both Natural and PAC commands.

### ADMIN Command - N and P

Entered from the SYSPAC library screen. ADMIN provides access to the PAC administrator functions if the user is authorized for those functions.

### DEACTIVATE Command - P

The DEACTIVATE command is used when you are displaying a specific object version and you need to deactivate and remove a Natural object from the PCF system file. Only the PAC administrator may use the DEACTIVATE command.

### MENU Command - N and P

Entered from the Logon Accepted screen to access the PAC Main Menu.

### PACADJST Command - N

The PACADJST command is entered from the SYSPACA library screen to invoke the File Adjust Function used to renumber PAC files without losing the integrity of the PAC system.

### REFRESH Command - P

The REFRESH direct command is valid for completed and backed-out events. It refreshes the current event to Pending state. The following information is removed from the refreshed event:

- Audit Report
- Authorization information
- Job information
- Schedule date

A refreshed event can be reused. PAC returns the message "PAC7337: (A) Migration event has been refreshed".

The REFRESH command may be used only on the Display Migration Event screen and by authorized users as per their User Profile settings.

### SYSPAC Command - N

Entered at the NEXT prompt on the Natural system library screen to access the PAC system. SYSPAC may alternatively be preceded by LOGON:

```
LOGON SYSPAC
```

### SYSPACA Command - N

Entered at the NEXT prompt on the Natural system library screen. The only object in SYSPACA is the menu object that logs the user on to SYSPAC and executes ADMIN. The SYSPACA object is provided for security reasons. ADMIN monitors whether the user is authorized to use SYSPACA.

### UNLOCK Command - P

The UNLOCK command invokes a screen for unlocking PAC data that was locked by a user or by a user's batch user ID. This is a PAC administrator function.

### Profile Command - N

The Natural Editor PROFILE command is available from the PAC command line. This enables you to utilize the majority of the editor profile functionality. Please refer to the Natural user documentation for more information.

## Commands to Set SYSPAC Options - P

Direct commands may be issued to SYSPAC or to Natural using the character string "//" as a prefix. These commands may be entered on the command line and issued from most PAC screens.

The following table lists and describes the special commands that may be issued to SYSPAC to set ON/OFF SYSPAC options and parameters:

| Command | Explanation |
|---|---|
| SET ON TRACK 9<br>SET OFF TRACK | This tool provides you with display checkpoints in order to aid in the debugging process.<br>To start it, enter SET ON TRACK 9.<br>To stop it, enter SET OFF TRACK. |

# Natural Program Editor Commands

The Natural program editor is used to edit various Natural objects.

This sectioncovers the following topics:

- Positioning Commands
- Edit Commands
- Line Commands

## Positioning Commands

| Cmd | Position . . . |
|---|---|
| +P, -P | forwards or backwards one page. |
| +, - | forwards or backwards one page. |
| +H, -H | forwards or backwards half a page. |
| T, -- | to top of program. |
| B, ++ | to bottom of program. |
| +nnnn | forwards "nnnn" lines (maximum 4 digits). |
| -nnnn | backward "nnnn" lines (maximum 4 digits). |
| nnnn | to line "nnnn". |
| X, Y | to the page containing the line marked with "X" or "Y". |

## Edit Commands

| Command | Description |
|---|---|
| ADD [(n)] | Adds "n" empty lines. If "n" is not specified, 9 lines (4 in split screen mode) will be added. |
| CHANGE | Scans for the value entered as "scandata" and replaces each such value with the value entered as "replacedata". The syntax for this command is: CHANGE `scandata'replacedata' Any special character which is not valid within a Natural variable name may be used as the delimiter character with the restriction that the same character must be used within any given command. |
| CLEAR | Clears the edit source work area. |
| DX | Deletes the X-marked line. |
| DY | Deletes the Y-marked line. |
| DX-Y | Deletes the block of lines delimited by X and Y. |
| EX | Deletes source lines from the top of the source area to, but not including, the X-marked line. |

| Command | Description |
|---|---|
| EY | Deletes source lines from the source line following the Y-marked line to the bottom of the source area. |
| EX-Y | Deletes all source lines in the source area excluding the block of lines delimited by X and Y. |
| POINT | Causes the line in which the ".N" line command was entered to be positioned to the top of the current screen. See also the line command ".P". |
| RESET | Deletes the current X and Y line markers and any marker previously set with the line command ".N". |
| SCAN [`scanvalue'] | Scans for data in the source area. SCAN (without parameter) will cause the SCAN menu to be produced. SCAN `scanvalue' will result in a scan for scan value. If the scan value is entered without delimiter characters, the entire character string which follows the keyword SCAN will be used as the scan value. |
| SCAN = | Causes a scan for the next occurrence of the scan value. The direction for a given scan command may also be explicitly specified by entering "SCAN =+" or "SCAN =-" prior to command execution. The first line which contains the scanned value is positioned to the top line after each SCAN= or SCAN+ command. Each line in which the scanned value is found will be marked with an "S". |
| SET ABS [ON\|OFF] | "SET ABS ON" causes the SCAN and CHANGE commands to operate in absolute mode, i.e., the value to be scanned/changed need not be delimited by blanks or special characters. "SET ABS OFF" (default) causes the SCAN and CHANGE commands to operate in non-absolute mode, i.e., the value to be scanned/changed must be delimited by blanks or special characters. |
| SET NUL [ON\|OFF] | "SET NUL ON" causes the value scanned with a SCAN command to be replaced by a null value. Default is OFF. |
| SHIFT [-\|+nn] | Shifts each source line delimited by the X and Y markers to the left or right. "nn" represents the number of characters the source line is to be shifted. Comment lines are not shifted. |
| SHIFT -- | Shifts each source line delimited by the X and Y markers to the leftmost position. Comment lines are not shifted. |
| SHIFT ++ | Shifts each source line delimited by the X and Y markers to the rightmost position (maximum 99 positions). Comment lines are not shifted. |
| * | Displays the edit command most recently entered. |
| *= | Causes the last command entered in the command line to be executed. |

### Line Commands

The notation "(nnnn)" indicates a repetition factor. The default repetition value is 1.

| Command | Description |
|---|---|
| .C(nnnn) | Copies the line in which the command was entered. |
| .CX(nnnn) | Copies the X-marked line. |
| .CY(nnnn) | Copies the Y-marked line. |
| .CX-Y(nnnn) | Copies the block of lines delimited by the X and Y markers. |
| .D(nnnn) | Delete line(s). The default is 1 line. |
| .I(n) | Adds "n" empty lines. "n" may be 1-9. If "n" is not specified, 9 lines (4 in split screen mode) will be added. |
| .J | Joins the current line with the next line. If the resulting line length is greater than the length of the editor screen line, the line will be marked with "L" and then must be separated again using the ".S" command before it can be modified. |
| .L | Will undo all modifications to the line since the last time was pressed. |
| .MX | Moves the X-marked line. |
| MY | Moves the Y-marked line. |
| .MX-Y | Moves the block of lines delimited by the X and Y markers. |
| .N | Marks a line as reference for the edit command POINT. |
| .P | Causes the line marked by this command to be positioned to the top of the screen. |
| .S | Splits the line at the position marked by the cursor. |
| .X | Marks the beginning of a block of lines to be processed. |
| .Y | Marks the end of a block of lines to be processed. |

# 2 Migration Utility

The PAC Migration utility may be used in conjunction with PAC migration events, using the command MIGLOAD.

MIGLOAD is used to load Natural cataloged objects (including cross-reference data if desired), saved objects, and views into the Natural system files.

**Note:** MIGLOAD is used to load objects into PAC that have been previously created in a PAC migration event (using workfile).

This chapter covers the following topics:

- **MIGLOAD**
- **User Exit for the Migration Utility**

# MIGLOAD

MIGLOAD is used to load Natural cataloged objects (and associated cross-reference data if available), saved objects, views, and Natural user error messages into the Natural system files.

MIGLOAD loads

- views into the Predict system file (FDIC);
- Natural programming objects and user error messages into the PAC system / Natural system files (FNAT and FUSER).
- Foreign objects.

Objects unloaded with either the NATUNLD or the TRANSFER utility are incompatible with the format supported by MIGLOAD and cannot be used.

The input for MIGLOAD is read from Natural Work File 1.

This sectioncovers the following topics:

- Accessing MIGLOAD
- Attributes of MIGLOAD
- Executing MIGLOAD Using Direct Commands
- Executing MIGLOAD in Batch Mode

- The MIGLOAD Report

### Accessing MIGLOAD

To access MIGLOAD, enter the command MIGLOAD at the NEXT prompt in Natural.

The Migrate Load Utility screen appears:

```
08:25:55            ***** PREDICT APPLICATION CONTROL *****         2001-04-25
User DBA                    - Migrate Load Utility -           Library SYSPAC




Processing type ..... L     ( 'L' to load or 'S' to scan )

File security exit .. N     ( User exit MIGEX003 must be used ? )


=========================================================================
 Load file ( NATURAL work file 1 ) must be created by PAC transfer event
=========================================================================
              PAC migrate load utility does not support
              load files created by NATUNLD utility
=========================================================================




Command ===>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit                                                   Canc
```

The following section describes the functions available on this screen. These may be used individually or in any combination.

### Attributes of MIGLOAD

The attributes of MIGLOAD displayed on the Migrate Load Utility screen are discussed in the following sections:

- Processing Type

- File Security Exit

#### Processing Type

The types of processing to be performed.

L Load objects for a library / libraries.

S Scan the MIGLOAD file without performing the load. The exact contents of the load file may be verified through the MIGLOAD report. By default, all objects on the load file are scanned.

#### File Security Exit

Whether or not the MIGEX003 security exit is to be used.

Y use MIGEX003 security exit.

N do not use MIGEX003 security exit.

### Executing MIGLOAD Using Direct Commands

Under the control of PAC, MIGLOAD is used specifically for loading Natural objects from a work file into and between statuses within PAC, and for loading objects into the PAA production environment. Outside the control of PAC, MIGLOAD may also be used for general migration.

Normally, objects unloaded from PAC are loaded using the following direct command:

```
MIGLOAD,LOAD
```

This section describes the variations of this command that can also be used.

The MIGLOAD direct command may be issued from most PAC screens, from Natural NEXT mode, or from the Natural Main Menu. The direct command syntax for loading Natural objects is as follows:

- Syntax 1

- Syntax 2

**Syntax 1**

```
MIGLOAD {LOAD | SCAN}
```

**Syntax 2**

```
MIGLOAD LOAD, {Y | N}
```

**Keywords Explanation**

| | |
|---|---|
| LOAD | Instructs PAC to load the contents of the work file depending on the selection criteria. |
| SCAN | Instructs PAC to scan, but not load, the contents of the work file. |

## Executing MIGLOAD in Batch Mode

Use a direct command to execute MIGLOAD in batch mode.

### Using Parameters

When executing MIGLOAD in batch mode using parameters, individual parameter values must be separated by the delimiter character (the default is ",").

It is recommended that the input-mode parameter (IM) be set to delimiter mode (IM=D).

### Using Direct Commands

Using direct commands, MIGLOAD is invoked in batch mode in the same way as online (see above); however, a direct command string must follow the MIGLOAD command.

You may enter direct commands in either of the following ways:

- The direct command string follows the MIGLOAD command on the same input line; each parameter in the command string may be delimited by a blank instead of the delimiter character.
- The direct command string follows the MIGLOAD command on the next input line; each parameter in the command string must be delimited by the delimiter character and not by a blank. If the direct command string is longer than one line, you must place the CF character at the end of the current line to be able to continue on the next line.

### Condition Codes in Batch Mode

If an error occurs during MIGLOAD execution, an error message appears and MIGLOAD terminates with a condition code.

## The MIGLOAD Report

MIGLOAD provides an audit report, which is a list of the objects that were loaded by MIGLOAD, including where they were loaded to. Additionally, after all MIGLOAD commands have been processed, a summary report is printed (displayed) giving totals for program objects loaded. The following screens are examples of the MIGLOAD report:

The following output (1-3) is an example of loading Natural objects.

Output 1

```
09:25:35            ***** PREDICT APPLICATION CONTROL *****         2001-04-25
User PACBAT                - Migrate Load Utility -           Library SYSPAC
                                                                   Page 1
            Processing load file created on 2001-04-25 at 09:22
            System files used for load:  NATURAL ... (00164,00247)
                                          PREDICT ... (00164,00235)
```

Output 2

```
09:25:35            ***** PREDICT APPLICATION CONTROL *****          2001-04-25
User PACBAT                 - Migrate Load Utility -           Library SYSPAC
                                                                    Page 2
Object       Object  Obj  Action          Date-Time     User      Dest.    NAT. Xrf
Name.Version  Type   Frm                 Save or Cat     ID      Library  Level Ind
GLOB1.0004    Global Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
GLOB2.0004    Global Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
GLOB3.0004    Global Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
GLOB4.0004    Global Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
GLOB5.0004    Global Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
MAP1.0004     Map    Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
MAP10.0004    Map    Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
MAP2.0004     Map    Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
MAP3.0004     Map    Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
MAP4.0004     Map    Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
MAP5.0004     Map    Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
MAP6.0004     Map    Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
MAP7.0004     Map    Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
MAP8.0004     Map    Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
MAP9.0004     Map    Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
SUB1.0004     Subpgm Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
SUB2.0004     Subpgm Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
SUB3.0004     Subpgm Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
SUB4.0004     Subpgm Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
SUB5.0004     Subpgm Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
SUB6.0004     Subpgm Cat Replacing      2001-04-04 10:42 DBA      DBA T23  3.1.4 ↩
S
```

Output 3

```
          *** MIGRATE Load has terminated successfully ***
 09:25:36            ***** PREDICT APPLICATION CONTROL *****         2001-04-25
 User PACBAT                - Migrate Load Utility -           Library SYSPAC
                                                                   Page 3
                     Statistical Report of Objects Processed
                                          Saved      Cataloged
                                       -----------  -----------
              Global Data Area .......            0            5
              Local/Param Data Area ..            0            0
              Maps ...................            0           10
              Helproutines ...........            0            0
              Subroutines ............            0            0
              Subprograms ............            0            8
              Programs ...............            0           24
              Copycode ...............            0
              Text ...................            0
              Process ................            0            0
              Miscellaneous Objects ..            0            0
              Views ..................            0
              Short Errors ...........            0
              Extended Errors ........            0
              Total programming obj ..            0           47
              Total objects Read .....            0           47
              Total Objects Loaded ...                        47 ***
NEXT FIN
NAT9995 NATURAL SESSION TERMINATED NORMALLY
```

The following output (1-3) is an example of loading foreign objects.

Output 1

```
 DATA LOAD
 10:49:20            ***** PREDICT APPLICATION CONTROL *****         2001-04-26
 User PACBAT                - Migrate Load Utility -               Library SYSPAC
                                                                   Page 1
             Processing load file created on 2001-04-26 at 10:47
             System files used for load:  NATURAL ... (00164,00247)
                                           PREDICT ... (00164,00235)
```

Output 2

```
10:49:20            ***** PREDICT APPLICATION CONTROL *****           2001-04-26
 User PACBAT                 - Migrate Load Utility -                Library SYSPAC
                                                                     Page 2
   Object     Object Obj  Action        Date-Time        User       Dest.    NAT. ↩
 Xrf
 Name.Version  Type   Frm                Save or Cat       ID       Library  Level ↩
 Ind
 BPPURG.0007   Module Cat Adding          2098-09-25 12:12   N/A
                 Destination Dataset ... DBA .PACDEMO.TST.JCL
 BR14.0007     Module Cat Replacing       2098-05-24 14:52   N/A
                 Destination Dataset ... DBA .PACDEMO.TST.JCL
 DEF164.0007   Module Cat Replacing       2098-05-25 15:38   N/A
                 Destination Dataset ... DBA .PACDEMO.TST.JCL
 FRM164.0007   Module Cat Replacing       2098-05-25 15:39   N/A
                 Destination Dataset ... DBA .PACDEMO.TST.JCL
 INPERR.0007   Module Cat Replacing       2099-04-04 20:39   N/A
                 Destination Dataset ... DBA .PACDEMO.TST.JCL
 INPL.0007     Module Cat Replacing       2098-05-20 16:01   N/A
                 Destination Dataset ... DBA .PACDEMO.TST.JCL
 INPLF23.0007  Module Cat Adding          2099-03-09 13:01   N/A
                 Destination Dataset ... DBA .PACDEMO.TST.JCL
 INPLN22.0007  Module Cat Adding          2098-05-20 10:52   N/A
                 Destination Dataset ... DBA .PACDEMO.TST.JCL
 INPL22.0007   Module Cat Adding          2099-04-08 10:22   N/A
                 Destination Dataset ... DBA .PACDEMO.TST.JCL
 INPL23.0007   Module Cat Adding          2098-06-17 23:19   N/A
                 Destination Dataset ... DBA .PACDEMO.TST.JCL
 INPL31.0007   Module Cat Adding          2099-03-09 16:54   N/A
                 Destination Dataset ... DBA .PACDEMO.TST.JCL
 LOADF.0007    Module Cat Adding          2099-04-08 10:08   N/A
                 Destination Dataset ... DBA .PACDEMO.TST.JCL
 NATLOAD.0007  Module Cat Adding          2098-08-19 15:17   N/A
                 Destination Dataset ... DBA .PACDEMO.TST.JCL
 NATUNLD.0007  Module Cat Adding          2098-08-20 09:14   N/A
                 Destination Dataset ... DBA .PACDEMO.TST.JCL
 ↩
```

Output 3

```
 10:49:46            ***** PREDICT APPLICATION CONTROL *****         2001-04-26
 User PACBAT                  - Migrate Load Utility -               Library SYSPAC
                                                                     Page 3
                     Statistical Report of Objects Processed
                                            Saved      Cataloged
                                        -----------  -----------
              Global Data Area .......            0            0
              Local/Param Data Area ..            0            0
              Maps ...................            0            0
              Helproutines ...........            0            0
              Subroutines ............            0            0
              Subprograms ............            0            0
              Programs ...............            0            0
              Copycode ...............            0
              Text ...................            0
              Process ................            0            0
              Miscellaneous Objects ..            0           33
              Views ..................            0
              Short Errors ...........            0
              Extended Errors ........            0
              Total programming obj ..            0           33
              Total objects Read .....            0           33
              Total Objects Loaded ...                        33 ***
NEXT FIN
NAT9995 NATURAL SESSION TERMINATED NORMALLY
```

The attributes of the MIGLOAD report screens are described in the following table:

| Attribute | Indicates . . . |
|---|---|
| Action | whether an existing object with the indicated name was replaced or not, or whether the object was added or deleted. |
| Date/Time of Save or Cat | the date and time the loaded object was saved or cataloged. |
| | Destination Dataset |
| | the name of the foreign object dataset to which the objects are being / were loaded. |
| Dest Library | the name of the library to which the object was loaded. |
| Object format | whether the loaded object is a saved or a cataloged object, or Xref data. |
| Object Name / Version | the name and version number of the object that was loaded. |
| Object Type | the type of the object that was loaded. |
| User ID | the user who saved or cataloged the loaded object. |
| Natural Level | the Natural system maintenance level of the loaded object. |
| Xref Ind | option specified in the load parameters for the disposition and conditions to be honored based on the presence of Xref data. |

## User Exit for the Migration Utility

PAC provides the MIGEX003 user exit for the migration utility.

The MIGEX003 user exit is an alias for PACEX003 and is invoked whenever PAC needs to access a Natural or Predict file to retrieve or store Natural or Predict objects.

Refer to the section **User Exits** for detailed information.

# 3 Compare Utility

With PAC Version 2.3.2 the Compare utility has been enhanced with a new user interface.

This chapter covers the following topics:

## Introduction

The Compare utility allows you to:

- Compare Natural objects outside of PAC control.
- Compare objects in two Natural libraries, outside of PAC control.
- Compare DDMs in a remote FDIC.
- Compare versioned objects (including Natural, Foreign, Error Messages and DDMs) in the PAC system.
- Compare objects under the control of PAA.
- Compare Natural objects in Production-controlled libraries.
- Manipulate the output from such comparisons for later use.

Depending on the environment from which the Compare utility is invoked, you access the relevant menu in either **Natural mode**, **PAA mode** or **PAC mode**.

If you want to use the Compare Utility in the way you used to do in earlier versions of PAC, you can do so using the **Extended Compare Utility** option from the Main Compare Menu.

## Using the Compare Utility

This section describes the features common to all Compare utility screens in Natural mode, PAC mode and PAA mode.

The following screen is an example of the Compare utility in Natural mode:

```
17:10:48           ***** Predict Application Control *****           2008-10-16
User HNO               - Compare Individual Objects -          Library SYSPAC
                                                                   Mode Natural
ORIGIN LOCATION
                                                 Srce or Loadable ... S (S/L)
DBnr ......... 164__  Fnr ... 54___  Password ..          Cipher ....
Library ...... SYSPAC___

Object ....... _________


DESTINATION LOCATION

DBnr ......... 164__  Fnr ... 54___  Password ..          Cipher ....
Library ...... SYSPAC___

Object ....... _________


Add. Criteria  N                                 Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp  Cmd               Libs  ErrM  DDMs  Clear  ↩
```

All Compare utility screens are divided into two sections, "Origin location" and "Destination location". The values you enter will be compared and the resultant data will be returned for viewing and, if necessary, further manipulation.

Some of the fields are pre-filled for your convenience. They may be overwritten.

The screen header displays the current time and date, the user, the name of the library from which you invoked the utility, and the functional compare mode.

From the main screen, you access the functions you need using the relevant menu codes or PF keys, as described later in this document.

For detailed descriptions relevant for each compare mode, see:

- **Natural Mode**
- **PAA Mode**
- **PAC Mode**

## Input Fields

The following input fields are the same in Natural mode, PAC mode and PAA mode:

| | |
|---|---|
| Object (Name) | Name of the objects to be compared. |
| Library | Name of the library in which the object is contained. |
| DBnr and Fnr | Database and file numbers of the objects to be compared. |
| Srce or Loadable | Source code or object code value. |
| Additional Criteria | Allows you to filter compared results. The default is N (no additional criteria). To activate the function, enter Y. |
| Output Options | Allows you to manipulate returned results. The default is N (no output options). To activate the function, enter Y. |

## PF Key Functions

The following general PF key functions are available:

| | | |
|---|---|---|
| PF1 | Help | Invokes the passive help screens. |
| PF2 | Menu | Invokes the main menu of the Compare utility. |
| PF3 | Exit | Exits the function. |
| PF4 | Crit | Invokes the additional criteria window for data filtering. |
| PF5 | Outp | Invokes the additional output options window for result manipulation. |
| PF12 | Clear | Clears the screen. |

### PF4 - Additional Criteria

This function is used to further enhance the selection criteria during the initial compare. It must be specified *before* the compare process is carried out.

When you press PF4, the Additional Criteria window offers you the following selections:

| | |
|---|---|
| Consider directory lines | When comparing, the Natural directory lines of the object will be included. |
| Ignore tail comments | When comparing, any line tail comments the object may have will be ignored. |
| Suppress comment lines | When comparing, any comment lines will be ignored / suppressed. |
| Discrepancies only | When the results are returned, only the discprepancies will be displayed. |

Enter Y to activate the selection.

### PF5 - Additional Output Options

This function is used to further manipulate the results of the compare process.

By default, the results are displayed on your screen. When you press PF5, the Additional Output Options window offers you the following output destinations:

| | |
|---|---|
| Display (Default=Y) | Displays the results on the screen. |
| Print | Sends results to a defined Natural printer. |
| Workfile | Sends results to Natural Workfile1. |
| Export | Invokes an additional window, that allows you to send the results to a Natural Text member in a Natural library. The data in this window is:<br><br>`Please select what results you would like to be exported.`<br><br>`Equal lines in 1st object ..... Y  Equal lines in 2nd object ↩ ..... Y`<br>`Different lines in 1st object ... Y  Different lines in 2nd ↩ object ... Y`<br>`No lines in 1st object ..... Y  Lines only in 2nd object ↩ ..... Y`<br>`Lines only in 1st object ..... Y  No lines in 2nd object ↩ ..... Y`<br><br>`DBnr 164  Fnr 247  Library .....  Object ↩` |
| Existing collation | Mark this option (Y), if the existing results are to be used for further munipulation. If not marked (N), the new results will be used. This is flagged as Y as soon as one compare has been carried out. |

This function can have effect both before and after the compare is carried out.

## Natural Mode

In Natural mode you can compare individual objects and lists of objects outside the control of a PAC environment.

This section covers the following topics:

- Accessing the Compare Utility in Natural Mode
- Comparing Individual Objects
- Comparing Natural Libraries or Lists of Objects
- Comparing Natural Error Messages

- Comparing Natural DDMs

### Accessing the Compare Utility in Natural Mode

The Compare utility interface for Natural mode is invoked, if you execute the COMPARE command from any Natural library other than SYSPAC or SYSPAA. You can use the Compare utility for any Natural library or Natural object you have access to.

### Comparing Individual Objects

This function allows you to compare a Natural object with any other Natural object.

When you access the Compare utility in Natural mode, the following screen is displayed:

```
17:10:48           ***** Predict Application Control *****          2008-10-16
User HNO               - Compare Individual Objects -          Library SYSPAC
                                                                   Mode Natural
ORIGIN LOCATION
                                                Srce or Loadable ... S (S/L)
DBnr ......... 164__  Fnr ... 54___  Password ..          Cipher ....
Library ...... SYSPAC___

Object ....... _________


DESTINATION LOCATION

DBnr ......... 164__  Fnr ... 54___  Password ..          Cipher ....
Library ...... SYSPAC___

Object ....... _________


Add. Criteria  N                                Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp  Cmd               Libs  ErrM  DDMs  Clear  ↩
```

This is the main screen for the Compare utility in Natural mode.

Fill the required fields Object Name, Library, DBnr / Fnr and Srce or Loadable in both sections with the data to be compared.

The values you enter will be compared and the resultant data will be returned for viewing and, if necessary, further manipulation.

The following Natural mode-specific input fields are available:

| | |
|---|---|
| Password | Allows you to provide a password that is used when accessing the FUSER without modifying MIGEX003 for each FUSER access. |
| Cipher | Allows you to provide a cipher that is used when accessing the FUSER without modifying MIGEX003 for each FUSER access. |

The following Natural mode-specific PF keys are available:

| | | |
|---|---|---|
| PF6 | Cmd | Invokes **direct command mode**. |
| PF9 | Libs | Invokes the Natural library / list compare function. |
| PF10 | ErrM | Invokes the Natural error message compare function. |
| PF11 | DDMs | Invokes the Natural DDM compare function. |

### Comparing Natural Libraries or Lists of Objects

This function allows you to compare a Natural library with any other Natural library, or a list of Natural objects with another list of Natural objects.

Enter PF9 to access the "Compare Natural Libraries" screen.

```
17:18:43           ***** Predict Application Control *****          2008-10-16
User HNO                - Compare Natural Libraries -          Library SYSPAC
                                                                   Mode Natural
ORIGIN LOCATION
                                                Srce or Loadable ... S (S/L)
DBnr ......... 164__  Fnr ... 54___  Password ..          Cipher ....
Library ...... SYSPAC___

Object ....... _________                        Object types  _________________


DESTINATION LOCATION
                                                Srce or Loadable ... S (S/L)
DBnr ......... 164__  Fnr ... 54___  Password ..          Cipher ....
Library ...... _________

Object ....... _________                        Object types  _________________


Add. Criteria  N                                Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp  Cmd         Nat         ErrM  DDMs  Clear  ↩
```

Fill the required fields DBnr / Fnr, Srce or Loadable and Library in both sections with the data to be compared. The other fields are optional.

The Object Types field is a multi-parameter field. It will accept more than one value at once, for example, to retrieve object types program as well as maps as well as local data areas, then you could enter P M L.

The values you enter will be compared and the resultant data will be returned for viewing and, if necessary, further manipulation.

The following Natural mode-specific PF keys are available:

| | | |
|---|---|---|
| PF6 | Cmd | Invokes **direct command mode**. |
| PF8 | Nat | Invokes the Natural individual compare function. |
| PF10 | ErrM | Invokes the Natural error message compare function. |
| PF11 | DDMs | Invokes the Natural DDM compare function. |

**Example**

The following description serves as an example when comparing one Natural library with another Natural library.

Example: Compare all map and program loadables in library JSYSPAC (164,11) with all map and program loadables in library FSYSPAC (164,11) and exporting the results to a workfile.

The following screen shows the parameter input:

```
17:20:44           ***** Predict Application Control *****           2008-10-16
User HNO                - Compare Natural Libraries -          Library SYSPAC
                                                                   Mode Natural
ORIGIN LOCATION
                                                Srce or Loadable ... S (S/L)
DBnr ......... 164__  Fnr ... 11___  Password ..          Cipher ....
Library ...... JSYSPAC__

Object ....... _________                        Object types  MP______________


DESTINATION LOCATION
                                                Srce or Loadable ... S (S/L)
DBnr ......... 164__  Fnr ... 11___  Password ..          Cipher ....
Library ...... FSYSPAC__

Object ....... _________                        Object types  MP______________


Add. Criteria  N                                Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp  Cmd         Nat         ErrM  DDMs  Clear  ↩
```

In the following screen the Output Options window is displayed. Workfile is selected.

```
17:20:44           ***** Predict Application Control *****          2008-10-16
User HNO                - Compare Natural Libraries -          Library SYSPAC
                                                                   Mode Natural
ORIGIN LOCATION
                                                Srce or Loadable ... S (S/L)
DBnr ......... 164__  Fnr ... 11___  Password ..          Cipher ....
Library ...... JSYSPAC__

Object ....... _________                        Object types  MP______________


DESTINATION LOCATION                             +------Output Options-------+
                                                 ! Display                Y  !
DBnr ......... 164__  Fnr ... 11___  Password ..  ! Print                  N  !
Library ...... FSYSPAC__                         ! Workfile               N  !
                                                 ! Export                 N  !
Object ....... _________                         ! Existing collation     N  !
                                                 +---------------------------+

Add. Criteria  N                                 Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp  Cmd         Nat         ErrM  DDMs  Clear  ↩
```

The following screen shows the result of the compare process:

```
There are 1030 semi-match and 157 discrepancy lines in this collation.
07:49:28               ***** PAC COMPARE Utility  *****              2008-10-16
Li  1M                       N    L      :2M                       N    L
    00164 00011 JSYSPAC                  :00164 00011 FSYSPAC
  > CFEPRERT P 2001-10-29 14:30
  <                                       CNVPAC22 P 2000-03-10 08:25
  > CNVPAPPL P 2001-10-29 14:04
  > CNVPAUDI P 2001-10-29 14:04
  > CNVPDDM  P 2002-04-17 11:17
  > CNVPDLOC P 2001-10-29 14:04
  > CNVPDLOL P 2001-10-29 14:04
  > CNVPDLOS P 2001-10-29 14:04
  > CNVPDPCF P 2001-10-29 14:04
  ! CNVPEVE  P 2001-10-29 14:04           CNVPEVE  P 2000-03-10 08:25
  ! CNVPFTT  P 2001-10-29 14:04           CNVPFTT  P 1998-12-08 11:20
  > CNVPFOFF P 2001-10-29 14:04
  > CNVPF000 P 2001-10-29 14:04
  ! CNVPISN  P 2001-10-29 14:04           CNVPISN  P 1998-05-28 19:27
  > CNVPNNL  P 2001-10-29 14:04
  ! CNVPRED  P 2001-10-29 14:04           CNVPRED  P 1998-05-28 19:27
  > CNVPREDP P 2001-10-29 14:04
  > CNVPVIEW P 2002-04-17 11:17
  > CNVPVK83 P 2001-10-29 14:04
Press ENTER to continue, enter '.' to stop:
```

The compare output has a summary line at the top of the page and shows the number of matches, semi-matches and discrepancies.

| | |
|---|---|
| ! | indicates that the object in list 1 has a timestamp different from that of its counterpart in list 2. |
| < | indicates that the object is present in list 2 but not in list 1. |
| > | indicates that the object is present in list 1 but not in list 2. |
| = | indicates that the timestamp of the objects coincide. |

In our example above the output was also sent to workfile1 for further manipulation.

### Comparing Natural Error Messages

This function allows you to compare a Natural error message with any other Natural error message, or a range of Natural error messages with another range of Natural error messages.

Enter PF10 to access the "Compare Natural Error Messages" screen.

```
15:48:34           ***** Predict Application Control *****          2002-04-25
User DBA               - Compare Natural Error Messages -        Library SYSPAC
                                                                   Mode Natural
ORIGIN LOCATION

DBnr ......... 164__  Fnr ... 31___
Library ...... DEMOPAC__

Message Number _____                            Language Code _


DESTINATION LOCATION

DBnr ......... 164__  Fnr ... 31___
Library ...... _________

Message Number _____                            Language Code _


Add. Criteria  N                                Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp  Cmd         Nat   Libs        DDMs  Clear
```

Fill the required fields Library and DBnr / Fnr in both sections with the data to be compared. The other fields are optional.

The values you enter will be compared and the resultant data will be returned for viewing and, if necessary, further manipulation.

The fields Message Number are relevant to determine wether you are comparing single error messages or lists of error messages. For example:

If you wanted to compare message number 0010 in library XYZ with message number 0010 in library ABC, then both the Message Number fields would have the value 0010 in them. However, if you wanted to compare all error messages in library XYZ beginning with 040 with all Error Messages in library ABC beginning with 040, then both Message Number fields would have the value 040* in them. This would compare a list of all 0400 messages with each other, i.e. 0400, 0401, 0402 etc.

The three controlling values that you can use in the Message Number field are:

| | |
|---|---|
| * | denotes the wildcard function. |
| < | denotes the less than function. |
| > | denotes the greater than function. |

Valid input examples for the Message Number field would be:

0010> all messages greater than 0010

0100* all messages starting with 100

0050< all messages less than 50

The following Natural mode-specific PF keys are available:

| | | |
|---|---|---|
| PF6 | Cmd | Invokes **direct command mode**. |
| PF8 | Nat | Invokes the Natural individual compare function. |
| PF9 | Libs | Invokes the Natural library / list compare function. |
| PF11 | DDMs | Invokes the Natural DDM compare function. |

### Comparing Natural DDMs

This function allows you to compare a Natural DDM with any other Natural DDM.

Enter PF11 to access the "Compare Natural DDMs" screen.

```
15:38:53           ***** Predict Application Control *****          2002-04-25
User DBA                  - Compare Natural DDMs -              Library SYSPAC
                                                                  Mode Natural
ORIGIN LOCATION

DBnr ......... 164__  Fnr ... 32___


Object ....... ________________________________


DESTINATION LOCATION

DBnr ......... 164__  Fnr ... 32___


Object ....... ________________________________


Add. Criteria  N                                Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp  Cmd         Nat   Libs  ErrM        Clear
```

Fill the required fields DBnr / Fnr and Object in both sections with the data to be compared. The other fields are optional.

The values you enter will be compared and the resultant data will be returned for viewing and, if necessary, further manipulation.

**Note:** For DDM comparisons the DBnr / Fnr must be a valid FDIC otherwise the following error message is returned: PAC6402: (E) The file is unavailable or unsuitable.

The following Natural mode-specific PF keys are available:

| | | |
|---|---|---|
| PF6 | Cmd | Invokes **direct command mode**. |
| PF8 | Nat | Invokes the Natural individual compare function. |
| PF9 | Libs | Invokes the Natural library / list compare function. |
| PF10 | ErrM | Invokes the Natural error message compare function. |

## PAA Mode

In PAA mode you can compare PAA-controlled objects, individual Natural objects and lists of objects outside the control of a PAA environment.

This section covers the following topics:

- Accessing the Compare Utility in PAA Mode
- Comparing Individual PAA Objects
- Comparing Natural Objects

### Accessing the Compare Utility in PAA Mode

The Compare utility interface for PAA mode is invoked, if you execute the COMPARE command from the library SYSPAA or choose option C from either the PAA Reporting main menu or the PAA Administration main menu. You can use the Compare utility for PAA-controlled objects which reside on the PAA system file.

When you access the Compare utility in PAA mode, the following screen is displayed:

```
11:53:16            ***** Predict Application Control *****         2002-04-25
User DBA                  - PAA Compare Main Menu -              Library SYSPAA
                                                                   Mode PAA

                       Code  Function or Mode
                       ----  ---------------------------------
                        A   Compare PAA Objects
                        N   Compare Natural Objects
                        X   Direct Command Mode
                        ?   Help
                        .   Exit









Command ===>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit                                                   Clear
```

Choose one of the Codes from the above menu to continue.

- **A to compare individual PAA objects**
- **N to compare individual Natural objects**
- **X to use direct command mode**

### Comparing Individual PAA Objects

This function allows you to compare a PAA-controlled object with any other PAA-controlled object.

Choose option A from the PAA Compare Main Menu to access the Compare Individual PAA Objects screen.

```
15:39:18           ***** Predict Application Control *****           2002-04-25
User DBA              - Compare Individual PAA Objects -         Library SYSPAC
                                                                       Mode PAA
ORIGIN LOCATION
FPAA DBnr .... 164__  Fnr ... 207__              Srce or Loadable ... S (S/L)
Location DBnr  164__  Fnr ... 31___              Library ........... _________
Application .. ___________________________________
Status ....... ___________________________________
Object ....... _________
PAA version .. _____  State _  Job __________    Current on .. __________ _____

DESTINATION LOCATION
FPAA DBnr .... 164__  Fnr ... 207__
Location DBnr  164__  Fnr ... 31___              Library ........... _________
Application .. ___________________________________
Status ....... ___________________________________
Object ....... _________
PAA version .. _____  State _  Job __________    Current on .. __________ _____

Add. Criteria  N                                 Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp                    List              Clear
```

Fill the required fields FPAA DBnr / Fnr, Srce or Loadable, Location DBnr / Fnr, Library and Object in both sections with the data to be compared. The other fields are optional.

The values you enter will be compared and the resultant data will be returned for viewing and, if necessary, further manipulation.

Optional fields are available for further selection criteria:

| | |
|---|---|
| Application | The name or alias of the PAA application. |
| Status | The name or alias of the PAA status. |
| PAA Version | The PAA version number of the object. (This differs from the PAC version number). |
| State | The PAA state the job is in.<br><br>S = Scheduled<br>C = Current<br>B = Backed up<br>H = Historical<br>R = Removed |
| Job | The PAA job number. |
| Current on | The date and time that the PAA object was made current. |

The following PAA mode-specific PF key is available:

| | | |
|---|---|---|
| PF9 | List | Invokes the PAA list compare function. |

### Comparing Lists of PAA Objects

This function allows you to compare a list of PAA-controlled objects with another list of PAA-controlled objects.

To compare lists of PAA objects, use the PF9 key from the "Compare Individual PAA Objects" menu.

```
15:41:28           ***** Predict Application Control *****           2002-04-25
User DBA               - Compare Lists of PAA Objects -          Library SYSPAC
                                                                       Mode PAA
ORIGIN LOCATION
FPAA DBnr .... 164__  Fnr ... 207__              Srce or Loadable ... S (S/L)
Location DBnr  164__  Fnr ... 31___              Library ........... _________
Application .. ___________________________________
Status ....... ___________________________________
Object ....... _________                         Object types  _________________


DESTINATION LOCATION
FPAA DBnr .... 164__  Fnr ... 207__              Srce or Loadable ... S (S/L)
Location DBnr  164__  Fnr ... 31___              Library ........... _________
Application .. ___________________________________
Status ....... ___________________________________
Object ....... _________                         Object types  _________________


Add. Criteria  N                                 Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp              PAA                     Clear
```

Fill the required fields FPAA DBnr / Fnr, Srce or Loadable, Location DBnr / Fnr and Library with the data to be compared. The other fields are optional.

The values you enter will be compared and the resultant data will be returned for viewing and, if necessary, further manipulation.

The following PAA mode-specific PF key is available:

| | | |
|---|---|---|
| PF8 | PAA | Invokes the PAA individual compare function. |

**Example**

The following is an example using the Compare utility in PAA mode to compare two Natural objects under the control of PAA.

Example: Compare all subprogram loadables in library LSYSPAC (164,11), application SYSPAC_222 with all subprogram loadables in library JSYSPAC (164,11), application SYSPAC_231 displaying results on the screen.

The following screen shows the parameter input:

```
08:27:12           ***** Predict Application Control *****          2002-04-30
User DBA              - Compare Lists of PAA Objects -         Library SYSPAC
                                                                        Mode PAA
ORIGIN LOCATION
FPAA DBnr .... 164__  Fnr ... 52___              Srce or Loadable ... L (S/L)
Location DBnr  164__  Fnr ... 11___              Library ............ LSYSPAC__
Application .. SYSPAC_222________________________
Status ....... ________________________________
Object ....... _________                         Object types  N________________


DESTINATION LOCATION
FPAA DBnr .... 164__  Fnr ... 52___              Srce or Loadable ... L (S/L)
Location DBnr  164__  Fnr ... 11___              Library ............ JSYSPAC__
Application .. SYSPAC_231________________________
Status ....... ________________________________
Object ....... _________                         Object types  N________________


Add. Criteria  N                                 Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp              PAA                     Clear
```

The following screen shows the result of the compare process:

```
There are 350 semi-match and 12 discrepancy lines in this collation.
08:30:30               ***** PAC COMPARE Utility  *****              2002-04-30
Li  1A                       NN   L 00000:2A                      NN   L 00000
    00164 00052 LSYSPAC  00164 00011     :00164 00052 JSYSPAC  00164 00011
  ! APINOBJV N 2000-04-13 16:49           APINOBJV N 2001-10-29 14:03
  ! APINOBLS N 1999-11-22 07:03           APINOBLS N 2001-10-29 14:03
  ! APINPATH N 1999-11-22 07:03           APINPATH N 2001-10-29 14:03
  ! APINPRF  N 1999-11-22 07:03           APINPRF  N 2001-10-29 14:03
  ! APINPRFS N 1999-11-22 07:03           APINPRFS N 2001-10-29 14:03
  ! APINSEL  N 1999-11-22 07:03           APINSEL  N 2001-10-29 14:03
  <                                       APINSELE N 2001-10-29 14:03
  ! APINSELO N 1999-11-22 07:03           APINSELO N 2001-10-29 14:03
  ! APINSRTN N 1999-11-22 07:03           APINSRTN N 2001-10-29 14:03
  ! APINSTAT N 1999-11-22 07:03           APINSTAT N 2001-10-29 14:03
  ! APINTEXT N 1999-11-22 07:03           APINTEXT N 2001-10-29 14:03
  ! AUTNMAPL N 2002-01-15 10:10           AUTNMAPL N 2002-01-09 23:38
  ! A15NCOPY N 1999-11-25 18:57           A15NCOPY N 2001-10-29 14:03
  ! A15NGLOB N 1999-11-25 18:57           A15NGLOB N 2001-10-29 14:03
  ! A15NPOPT N 1999-11-25 18:57           A15NPOPT N 2001-10-29 14:03
  ! A15NVAL1 N 1999-11-25 18:57           A15NVAL1 N 2001-10-29 14:03
  ! CFENAUXD N 2000-02-20 14:05           CFENAUXD N 2001-10-29 14:30
  ! CFENAUXS N 2000-02-20 14:05           CFENAUXS N 2001-10-29 14:30
  ! CFENCMP  N 2000-02-20 14:05           CFENCMP  N 2001-10-29 14:30
Press ENTER to continue, enter '.' to stop:
```

### Comparing Natural Objects

This function allows you to compare Natural objects with other Natural objects and Natural libraries with other Natural libraries as described in the following sections:

- **Comparing Individual Objects**
- **Comparing Natural Libraries or Lists of Objects**

Starting the comparison from PAA instead of under Natural allows you to compare Natural objects outside of the PAA environment (Production), as these objects / libraries may not be accessible from their development environment.

Choose option N from the PAA Compare Main Menu to access the **Compare Individual Objects** screen.

# PAC Mode

In PAC mode you can compare PAC-controlled objects as well as individual Natural objects and lists of objects outside the control of a PAC environment.

This section covers the following topics:

- Accessing the Compare Utility in PAC Mode
- Comparing Versioned Objects
- Comparing Natural Objects
- Comparing PAA Objects

## Accessing the Compare Utility in PAC Mode

The Compare utility interface for PAC mode is invoked, if you execute the COMPARE command from the library SYSPAC or using option C in the PAC Reporting main menu. You can use the Compare utility for PAC-controlled objects which reside on the PAC system file.

When you access the Compare utility in PAC mode, the following screen is displayed:

```
15:42:12           ***** Predict Application Control *****          2002-04-25
User DBA                 - PAC Compare Main Menu -               Library SYSPAC
                                                                       Mode PAC


                   Code  Function or Mode
                   ----  --------------------------------
                    V    Compare Versioned Objects
                    N    Compare Natural Objects
                    A    Compare PAA Objects
                    X    Direct Command Mode
                    E    Extended Compare
                    ?    Help
                    .    Exit

            Code .. _




Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit                                                   Canc
```

Choose one of the Codes from the above menu to continue.

- **V to compare versioned objects**

- **N to compare individual Natural objects**
- **A to compare individual PAA objects**
- **X to use direct command mode**
- **E to use extended compare**

### Comparing Versioned Objects

This function allows you to compare a PAC-controlled Natural object with any other PAC-controlled Natural object.

Choose option V from the PAC Compare Main Menu to access the "Compare Versioned Natural Objects" screen.

```
15:42:40           ***** Predict Application Control *****          2002-04-25
User DBA           - PAC Compare Versioned Natural Objects -    Library SYSPAC
                                                                       Mode PAC
ORIGIN LOCATION
ACF DBnr ..... 164__  Fnr ... 205__              Srce or Loadable ... S (S/L)

Application .. ________________________________
Status ....... ________________________________
Object ....... _________
Version Nr ... _____                             Current on .. __________ _____

DESTINATION LOCATION
ACF DBnr ..... 164__  Fnr ... 205__

Application .. ________________________________
Status ....... ________________________________
Object ....... _________
Version Nr ... _____                             Current on .. __________ _____

Add. Criteria  N                                 Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp        Forei       List  ErrM  DDMs  Clear
```

Fill the required fields ACF DBnr / Fnr, Srce or Loadable, Application and Object in both sections with the data to be compared. The other fields are optional.

The values you enter will be compared and the resultant data will be returned for viewing and, if necessary, further manipulation.

Optional fields are available for further selection criteria:

| | |
|---|---|
| Application | The name of the PAC application. |
| Status | The name of the PAC status. |
| Version Nr | The PAC version number of the object. |
| Current on | The date and time that the PAC object was made current. |

The following PAC mode-specific PF keys are available:

| | | |
|---|---|---|
| PF7 | Forei | Invokes the versioned foreign objects compare function. |
| PF8 | Nat | Invokes the Natural versioned objects function |
| PF9 | List | Invokes the lists of versioned objects compare function. |
| PF10 | ErrM | Invokes the versioned error message compare function. |
| PF11 | DDMs | Invokes the versioned DDMs compare function. |

#### Compare Versioned Foreign Objects

This function allows you to compare a PAC-controlled foreign object with any other PAC-controlled foreign object.

To compare versioned foreign objects, use the PF7 key from the "PAC Compare Versioned Natural Objects" menu.

```
15:43:08           ***** Predict Application Control *****          2002-04-25
User DBA           - PAC Compare Versioned Foreign Objects -     Library SYSPAC
                                                                        Mode PAC
ORIGIN LOCATION
ACF DBnr ..... 164__  Fnr ... 205__              Srce or Loadable ... S (S/L)

Application .. ____________________________________
Status ....... ____________________________________
Object ....... ____________________________________ Object type . ____
Version Nr ... _____                               Current on .. __________ _____

DESTINATION LOCATION
ACF DBnr ..... 164__  Fnr ... 205__

Application .. ____________________________________
Status ....... ____________________________________
Object ....... ____________________________________ Object type . ____
Version Nr ... _____                               Current on .. __________ _____

Add. Criteria  N                                   Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp              Nat   List  ErrM  DDMs  Clear
```

Fill the required fields ACF DBnr / Fnr, Srce or Loadable, Application, Object and Object type in both sections with the data to be compared. The other fields are optional.

The values you enter will be compared and the resultant data will be returned for viewing and, if necessary, further manipulation.

The Object type field is unique to PAC versioned foreign objects and is a 3 letter user-defined value, for example JCL.

#### Compare Lists of Versioned Objects

This function allows you to compare a list of PAC-controlled objects with another list of PAC-controlled objects.

To compare lists of versioned objects, use the PF9 key from the "PAC Compare Versioned Natural Objects" menu.

```
15:43:33           ***** Predict Application Control *****           2002-04-25
User DBA           - PAC Compare List of Versioned Objects -     Library SYSPAC
                                                                       Mode PAC
ORIGIN LOCATION
ACF DBnr ..... 164__  Fnr ... 205__              Srce or Loadable ... S (S/L)

Application .. ___________________________________
Status ....... ___________________________________
Object ....... _________                         Object types  ________________
Version Nr ... _____                             Current on .. __________ _____

DESTINATION LOCATION
ACF DBnr ..... 164__  Fnr ... 205__              Srce or Loadable ... S (S/L)

Application .. ___________________________________
Status ....... ___________________________________
Object ....... _________                         Object types  ________________
Version Nr ... _____                             Current on .. __________ _____

Add. Criteria  N                                 Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp        Forei Nat         ErrM  DDMs  Clear
```

Fill the required fields ACF DBnr / Fnr, Srce or Loadable and Application in both sections with the data to be compared. The other fields are optional.

The values you enter will be compared and the resultant data will be returned for viewing and, if necessary, further manipulation.

### Compare Versioned Error Messages

This function allows you to compare one or more PAC-controlled versioned error message with any other PAC-controlled versioned error message(s).

To compare versioned error messages, use the PF10 key from the "PAC Compare Versioned Natural Objects" menu.

```
15:43:57           ***** Predict Application Control *****          2002-04-25
User DBA           - PAC Compare Versioned Error Messages -      Library SYSPAC
                                                                       Mode PAC
ORIGIN LOCATION
ACF DBnr ..... 164__  Fnr ... 205__

Application .. ________________________________
Status ....... ________________________________
Message Number _____                             Language Code _
Version Nr ... _____                             Current on .. __________ _____

DESTINATION LOCATION
ACF DBnr ..... 164__  Fnr ... 205__

Application .. ________________________________
Status ....... ________________________________
Message Number _____                             Language Code _
Version Nr ... _____                             Current on .. __________ _____

Add. Criteria  N                                 Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp        Forei Nat   List        DDMs  Clear
```

Fill the required fields ACF DBnr / Fnr and Application in both sections with the data to be compared. The other fields are optional.

The values you enter will be compared and the resultant data will be returned for viewing and, if necessary, further manipulation.

### Compare Versioned DDMs

This function allows you to compare a PAC-controlled versioned DDM with any other PAC-controlled versioned DDM.

To compare versioned DDMs, use the PF11 key from the "PAC Compare Versioned Natural Objects" menu.

```
15:44:19           ***** Predict Application Control *****           2002-04-25
User DBA                - PAC Compare Versioned DDMs -            Library SYSPAC
                                                                        Mode PAC
ORIGIN LOCATION
ACF DBnr ..... 164__  Fnr ... 205__

Application .. ________________________________
Status ....... ________________________________
Object ....... ________________________________
Version Nr ... _____                             Current on .. __________ _____

DESTINATION LOCATION
ACF DBnr ..... 164__  Fnr ... 205__

Application .. ________________________________
Status ....... ________________________________
Object ....... ________________________________
Version Nr ... _____                             Current on .. __________ _____

Add. Criteria  N                                 Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp        Forei Nat   List  ErrM        Clear
```

Fill the required fields ACF DBnr / Fnr, Application (Predict applications only) and Object in both sections with the data to be compared. The other fields are optional. The values you enter will be compared and the resultant data will be returned for viewing and, if necessary, further manipulation.

Example

The following is an example using the Compare utility in PAC mode to compare two versioned DDMs.

Example: Compare a versioned DDM named load-module version 1 in application Predict with that of the same named DDM version 1 in application DDMS_FOR_231.

The following screen shows the parameter input:

```
08:59:01           ***** Predict Application Control *****          2002-04-30
User DBA               - PAC Compare Versioned DDMs -          Library SYSPAC
                                                                         Mode PAC
ORIGIN LOCATION
ACF DBnr ..... 164__  Fnr ... 50___

Application .. PREDICT___________________________
Status ....... ________________________________
Object ....... LOAD-MODULE_______________________
Version Nr ... 1____                             Current on .. __________ _____

DESTINATION LOCATION
ACF DBnr ..... 164__  Fnr ... 50___

Application .. DDMS_FOR_231______________________
Status ....... ________________________________
Object ....... LOAD-MODULE_______________________
Version Nr ... 1____                             Current on .. __________ _____

Add. Criteria  N                                 Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp        Forei Nat   List  ErrM        Clear
```

The following screen shows the result of the compare process:

```
There are 34 match and 125 discrepancy lines in this collation.
09:00:05               ***** PAC COMPARE Utility  *****              2002-04-30
Sq  1C LOAD-MODULE           D      00001:2C LOAD-MODULE           D      00001
    00164 00050 PREDICT                  :00164 00050 DDMS_FOR_231
  = 0001 DB: 148 FILE: 044  - LOAD-MODULE 0001 DB: 148 FILE: 044  - LOAD-MODULE
  ! 0002                                  0002                                Ca
  = 0003 T L DB Name                      0003 T L DB Name
  = 0004 - - -- ------------------------- 0004 - - -- -------------------------
  ! 0005 *       Generation started      0005 *      Generation started
  ! 0006 *       at 95-06-27 11:14:42    0006 *      at 97-12-19 11:45:18
  ! 0007 *       by user ESY211          0007 *      by user ESY213
  = 0008 *                               0008 *
  ! 0009 * *                             0009 *
  ! 000A * *    Fields common to all Op-s 000A *      Fields common to all Op-s
  ! 000B * *                             000B *
  = 000C   1 EC ERROR-CODE                000C   1 EC ERROR-CODE
  ! 000D *           Message code.       000D *      Message code.
  = 000E   1 ET ERROR-TEXT                000E   1 ET ERROR-TEXT
  ! 000F *           Text describing mess 000F *      Text describing message.
  = 0010   1 DB NODE                      0010   1 DB NODE
  ! 0011 *           Node ID (DBID).     0011 *      Node ID (DBID).
  = 0012   1 ND NODE-NAME                 0012   1 ND NODE-NAME
  ! 0013 * *                             0013 *
Press ENTER to continue, enter '.' to stop:
```

### Comparing Natural Objects

This function allows you to compare Natural objects with other Natural objects and Natural libraries with other Natural libraries.

Choose option N from the PAC Compare Main Menu to access the "Compare Individual Objects" screen.

Starting the comparison from PAC instead of under Natural allows you to have the full functions of the utility available in a central place.

```
17:10:48           ***** Predict Application Control *****           2008-10-16
User HNO               - Compare Individual Objects -          Library SYSPAC
                                                                   Mode Natural
ORIGIN LOCATION
                                                Srce or Loadable ... S (S/L)
DBnr ......... 164__  Fnr ... 54___  Password ..          Cipher ....
Library ...... SYSPAC___

Object ....... _________


DESTINATION LOCATION

DBnr ......... 164__  Fnr ... 54___  Password ..          Cipher ....
Library ...... SYSPAC___

Object ....... _________


Add. Criteria  N                                Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp  Cmd               Libs  ErrM  DDMs  Clear  ↩
```

For detailed description, see the section **Natural Mode**.

### Comparing PAA Objects

This function allows you to compare single PAA-controlled objects with other PAA-controlled objects and lists of PAA-controlled objects with other lists of PAA-controlled objects.

Choose option A from the PAC Compare Main Menu to access the "Compare Individual PAA Objects" screen.

Starting the comparison from PAC instead of under PAA allows you to have the full functions of the utility available in a central place.

Please note that in order to use this function, you need access to your FPAA file from your PAC environment.

```
15:45:39           ***** Predict Application Control *****          2002-04-25
User DBA              - Compare Individual PAA Objects -       Library SYSPAC
                                                                        Mode PAA
ORIGIN LOCATION
FPAA DBnr .... 164__  Fnr ... 207__              Srce or Loadable ... S (S/L)
Location DBnr  164__  Fnr ... 31___              Library ........... _________
Application .. ___________________________________
Status ....... ___________________________________
Object ....... _________
PAA version .. _____  State _  Job __________    Current on .. __________ _____

DESTINATION LOCATION
FPAA DBnr .... 164__  Fnr ... 207__
Location DBnr  164__  Fnr ... 31___              Library ........... _________
Application .. ___________________________________
Status ....... ___________________________________
Object ....... _________
PAA version .. _____  State _  Job __________    Current on .. __________ _____

Add. Criteria  N                                 Output Options ..... N
Command =====> ____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Crit  Outp                    List              Clear
```

## Direct Command Mode

When you invoke the PAC compare direct command mode, the following screen is displayed:

```
09:48:58          ***** Predict Application Control *****          2002-05-13
User DBA            - PAC Compare Direct Command Mode -       Library SYSPAC



Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  Exit
```

You can invoke the direct command mode online by either using menu option "X" or, when in PAC mode, with the PF6 key (Cmd). A blank screen is displayed in which you enter your direct commands. If the length of the command exceeds the screen size, you can use the continuation character (%), see the COMPARE command examples **COMPARE Command Examples**. However, in order to obtain the best possible results from the Compare Utility, it is recommended that you use the direct command mode option in batch.

The following points are important and worth noting when using the direct command mode.

- the first command must always be COMPARE;
- commas are required to separate syntax elements (see examples later in this section);
- APPLICATION can no longer be abbreviated as APPL;
- LIB or LIBRARY is a mandatory entry;
- all commands need a WHERE-clause indicating DBnr's and Fnr's. (Previously if this was not specified, the Compare utility assumed the current settings of the ACF / FUSER).

| Keywords | Explanation |
|---|---|
| ALL | Selects only saved components for processing. |
| CAT | Selects only cataloged components for processing. (Any corresponding saved components will not be processed.) |
| SAVED | Selects all saved components for processing. (Any corresponding cataloged object will not be processed.) |

| Keywords | Explanation |
|---|---|
| STOWED | Selects all saved components for processing. (Any corresponding cataloged object will not be processed.) |
| IN/FM | Introduces the specification of the first list or object to be compared. |
| TO | Introduces the specification of the second list or object to be compared. |
| APPLICATION | Name of a PAC application that contains the object. |
| LIBRARY | Name of the library where the object exists. Mandatory. |

- COMPARE, object-type, and object-name1 must normally be the first three parameters of the command string.
- A period mark indicates the end of command. If a "." is detected anywhere within a command string, all data after the "." will be ignored.
- The library-name must be specified immediately after the IN, FM, and TO keywords.
- The application-name must follow the APPLICATION keyword, which must be specified immediately after the IN, FM, and TO keywords.
- The where-clause1 must always follow the IN or FM keyword and library-name1 or application-name1; however, the keywords and values within the clause may be specified in any order.
- The where-clause2 must always follow the TO keyword and library-name2 or application-name2; however, the keywords and values within the clause may be specified in any order.
- The keywords and values of the with-clause may be specified in any order; and the with-clause may be placed anywhere within the direct command string, except in the first three positions.

### WHERE-Clause

The where-clause is mandatory. It specifies the location of the library specified in the IN/FM (from) or TO clauses.

Note that the keywords and values within the clause may be specified in any order.

| Keywords | Explanation |
|---|---|
| WHERE | Indicates the start of a where-clause. |
| CIPHER | Cipher code of the file. (Ignored). |
| DBID | Database number of the specified library. |
| FNR | File number of the specified library. |
| NAME | VSAM name (FCT) of the file (VSAM only). (Ignored). |
| PASSWORD | Password of the file. (Ignored). |
| SEC | DBnr, Fnr, password, cipher of the Natural Security file against which security validation must be performed. (Ignored). |
| STATUS | Name of the status. |
| VERSION | Version number of an/the object(s) to be compared. |

### WITH-Clause

The with-clause is optional. It specifies attributes of the objects to be selected.

Note that the keywords and values of the with-clause may be specified in any order; and the with-clause may be placed in any location within the direct command string, except in the first three positions.

| Keywords | Explanation |
|---|---|
| WITH | (Optional) Indicates the start of a with-clause. |
| DIFFER | Selects for processing all programming objects with different saved and cataloged dates. (Ignored). |
| EXPAND | Expands the compare report to show a comparison of both directories and source code; a short compare list is not displayed. The source comparison is written to CM-PRT01. |
| FMDATE | Limits the selection by the save/catalog date. |
| FMTIME | Limits the selection by the save/catalog time. |
| MON | Activates online trace facility (not valid in batch). (Ignored). |
| SHORT | Selects for processing all programming objects found in the origin library. If SHORT is not specified, all objects in both libraries will be processed. (Ignored). |
| TID | Identifies the terminal ID where the saved/cataloged objects must have been processed to be selected. |
| TODATE | Identifies the save/catalog date after which no objects are selected. |
| TOTIME | Identifies the save/catalog time after which no objects are selected. |
| TYPE | Identifies the type(s) of objects to be selected. |
| USER | Identifies the ID of the user who saved/cataloged the objects to be selected for processing. |

### COMPARE Command Examples

To compare the saved components of a particular object in two different Natural libraries.:

```
COMPARE
COMPARE,ALL,,TRNSTOWP,FM,LIB,ASYSPAC,WHERE,DBID,164,FNR,11,%
TO,LIB,PAC14CSC,WHERE,DBID,164,FNR,11
.
```

To compare the cataloged component of a particular object in two different Natural libraries:

```
COMPARE
COMPARE,CATALOGED,,TRNSTOWP,FM,LIB,ASYSPAC,WHERE,DBID,164,FNR,11,%
TO,LIB,PAC14CSC,WHERE,DBID,164,FNR,11
.
```

To compare the saved component of a particular object in two different Natural libraries:

```
COMPARE
COMPARE,SAVED,,TRNSTOWP,FM,LIB,ASYSPAC,WHERE,DBID,164,FNR,11,%
TO,LIB,PAC14CSC,WHERE,DBID,164,FNR,11
.
```

To compare the saved component of a particular object in two different Natural libraries:

```
COMPARE
COMPARE,STOWED,,TRNSTOWP,FM,LIB,ASYSPAC,WHERE,DBID,164,FNR,11,%
TO,LIB,PAC14CSC,WHERE,DBID,164,FNR,11
.
```

To compare a range of components in two different Natural libraries:

```
COMPARE
COMPARE,ALL,,TRN*,FM,LIB,ASYSPAC,WHERE,DBID,164,FNR,11,%
TO,LIB,PAC14CSC,WHERE,DBID,164,FNR,11
.
```

To compare a component of an application in two different PAC statuses with different saved and cataloged dates:

```
COMPARE
COMPARE,ALL,,TRNSTOWP,IN,APPLICATION,SYSPAC_214,STATUS,CONTROL,%
WHERE,DBID,164,FNR,50,%
TO,APPLICATION,SYSPAC_214,WHERE,STATUS,PRODUCTION,%
WHERE,DBID,164,FNR,50,WITH,DIFF
.
```

To compare a range of components of an application in two different PAC statuses:

```
COMPARE
COMPARE,ALL,,TRN*,IN,APPLICATION,SYSPAC_214,STATUS,CONTROL,%
WHERE,DBID,164,FNR,50,%
TO,APPLICATION,SYSPAC_214,WHERE,STATUS,PRODUCTION,%
WHERE,DBID,164,FNR,50,WITH,DIFF
.
```

To compare a component of an application with two different versions:

```
COMPARE
COMPARE,ALL,,TRNSTOWP,IN,APPLICATION,SYSPAC_214,VERSION,2,%
WHERE,DBID,164,FNR,50,%
TO,APPLICATION,SYSPAC_214,WHERE,VERSION,3,%
```

```
WHERE,DBID,164,FNR,50
.
```

To compare a range of components of an application with two different versions:

```
COMPARE
COMPARE,ALL,,TRN*,IN,APPLICATION,SYSPAC_214,VERSION,2,%
WHERE,DBID,164,FNR,50,%
TO,APPLICATION,SYSPAC_214,WHERE,VERSION,3,%
WHERE,DBID,164,FNR,50
.
```

To compare a component of an application with two different versions, using the parameter expand:

```
COMPARE
COMPARE,ALL,,MENU,IN,APPLICATION,SYSPAC_214,VERSION,1,%
WHERE,DBID,164,FNR,50,%
TO,APPLICATION,SYSPAC_222,WHERE,VERSION,1,%
WHERE,DBID,164,FNR,50,WITH,EXPAND
.
```

## Extended Compare Utility

The Extended Compare Utility is the old Compare utility, that was delivered with PAC Version 2.2. It was thought necessary to leave this functionality in the product for those users who have already mastered the more complex functions that this has provided.

This section covers the following topics:

- Redefine Scratch Pad
- Define List or Sequence
- Operate on Lists
- Operate on Sequences
- Operate on Collations
- Compare Adhoc
- Extended Direct Command Mode
- The former COMPARE Command Syntax

Choose option E from the PAC Compare Main Menu to access the "Compare Utility Extended Menu" screen.

```
15:29:25           ***** PREDICT APPLICATION CONTROL *****          2007-05-25
User POR                - COMPARE Utility Main Menu -          Library SYSPAC


                   Code  Function or Mode
                   ----  --------------------------------
                     P   Redefine Scratch Pad
                     D   Define List or Sequence
                     L   Operate on Lists
                     S   Operate on Sequences
                     C   Operate on Collations
                     A   Compare Ad Hoc
                     M   Direct Command Mode
                     O   Close Printer
                     ?   Help
                     .   Exit
                   ----  --------------------------------
             Code ..


Pad 164     166   Lists          Seqs          Colls          Reps
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  Exit                                                   Canc
```

PF Keys available from this panel are PF3 (Exit or Return) and PF12 (Cancel).

The short code indicators in the panel at the bottom are:

| | |
|---|---|
| Pad | Current DBnr and Fnr for scratch pad settings. Default: PCF settings. |
| Lists | Number of each list defined for this session. Blank spaces are displayed for lists which were not defined as shown in the screen in section **Operate on Lists**.<br>A maximum of 8 lists can be defined. |
| Seqs | Number of each sequence defined for this session. Blank spaces are displayed for sequences which were not defined as shown in the screen in section **Operate on Lists**.<br>A maximum of 8 sequences can be defined. |
| Colls | Number of each collation defined for this session. Blank spaces are displayed for collations which were not defined as shown in the screen in section **Operate on Lists**.<br>A maximum of 8 collations can be defined. |
| Reps | Number of each report defined for this session. Blank spaces are displayed for reports which were not defined.<br>A maximum of 8 reports can be defined. |

### Redefine Scratch Pad

Enter function code P in the Compare Utility Main Menu to define / modify the default values set for the scratch pad. If the file definition is omitted, PCF is assumed.

The scratch pad is used to temporarily store lists and definitions of sequences used during your session. The current value of this parameter is displayed on the Main Menu panel on the left side.

### Define List or Sequence

Enter function code D in the Compare Utility Main Menu to either define a list (of common, PAC or PAA objects) for mass compare, or a sequence (individual compare) for later use in the Compare utility.

Before starting the actual compare process, you need to define lists or sequences using various selection criteria.

The Compare Utility Entity Definition screen appears:

```
15:38:26           ***** PREDICT APPLICATION CONTROL *****          2000-02-29
User DBA            - COMPARE Utility Entity Definition -        Library SYSPAC

     Entity............ _ (L,S)     Kind............... _ (coMmon,paC,paA,Event)
     Number............ _ (1..8)    Object Group...... _ (N,E,D,F,M)









Pad 164     135   Lists          Seqs          Colls          Reps
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  Retr                                                   Canc
```

The following parameters are available:

| Parameter | Description |
|---|---|
| Entity | List or Sequence |
| Kind | "M" Common Objects outside of PAC control<br>"C" PAC objects under the control of PAC<br>"A" PAA objects under the control of PAA<br>"E" PAC events |
| Number | The number assigned to the defined list or sequence (1 to 8) |
| Object Group | "N" Natural objects<br>"E" Error Messages<br>"D" Data Definition Modules<br>"F" Foreign Objects<br>"M" Migration lists |

Enter the five values and press ENTER to continue. Depending on your input various screens are displayed in which you can specify the attributes for defining lists or sequences.

The following screen shows an example of the definition of a list (number 3) of common Natural modules in library KSYSPAC.

```
15:40:52           ***** PREDICT APPLICATION CONTROL *****          2000-02-29
User DBA            - COMPARE Utility Entity Definition -        Library SYSPAC

     Entity............ l (L,S)     Kind............... m (coMmon,paC,paA,Event)
     Number............ 3 (1..8)    Object Group...... n (N,E,D,F,M)

     Definition of list 3
     Database number... 164__       Src/Loadable...... l (S/L)
     File number....... 11___       Object types...... p_______________

     Library........... ksyspac_

     Object............ ___________________________________

     Time from......... YYYY-MM-DD HH:II
     Time to........... YYYY-MM-DD HH:II
     User.............. ________


Pad 164     135   Lists   3      Seqs          Colls          Reps
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  Retr                                                   Canc
```

### Attributes for Defining Lists and Sequences

| | |
|---|---|
| Database number | Database number of the origin environment where the object / objects exist. |
| File number | File number of the origin environment where the object/objects exist. For Common objects the Fnr must be a valid FUSER/FNAT/FDIC file. For PAC objects the Fnr must be a valid ACF file.<br>For PAA objects the Fnr must be a valid PAA file. |
| Src/Loadable | Type of objects; either source code or loadable code. |
| Object type | The type(s) of objects to be compared. For Common objects please see valid types in the PAC User's Guide. For foreign objects this type value is user defined. |
| Library | The name of the library where the objects reside. |
| Object | The name of the object to be compared. |
| Time from | The time stamp from of the object selected.<br>Format is YYYY-MM-DD HH:II |
| Time to | The time stamp to of the object selected.<br>Format is YYYY-MM-DD HH:II |
| User | The valid user ID of the user that saved / stowed / compiled the object. |

The following attributes are available depending on the combination of the attributes Kind and Object Group you specified:

| | |
|---|---|
| Ebyte | Used for object group (E) (error messages) only. If this field is left blank then all short error messages will be returned. If this field is populated with a valid Natural language code, then the long error messages for that language code will be returned. |
| Application | The PAC application of the object(s) to be defined. This field is mandatory for definition of kind PAC. |
| Status | The PAC status of the object(s) to be defined. |
| Event | If status is either blank or Control it refers to an immigration event. Anything else refers to an emigration event. |
| PAC version | The object version number assigned by PAC. |
| Current on | For a PAC definition this will return the most recent objects on the value specified. For PAA definitions then it will return objects that will be equal to the PAA field `current on'. |
| | The following six location attributes are used only when Kind=A (PAA). |
| Database number | PAA location database number. |
| File number | PAA location file number (FUSER/FNAT/FDIC) |
| Library | PAA location library (valid FUSER/FNAT library) |
| Node | PAA location ESY node for foreign objects. |
| Volume or BS2 type | PAA location Volume or BS2 type for foreign objects. |
| Data set | PAA location dataset name. |
| PAA version | The object version number assigned by PAA. |

| | |
|---|---|
| State | The PAA value of `state'. This can be Backed out, Current, Historical, Removed, and Scheduled. |
| Job | The number of the PAA job that was used to migrate the object into PAA. |

The result of this list definition is returned to the Compare Utility Main Menu displaying a short text on top of the screen of the number of objects that the definition obtained, as seen in the following screen.

```
15:28:16           ***** PREDICT APPLICATION CONTROL *****           2007-05-25
User POR                - COMPARE Utility Main Menu -          Library SYSPAC


                   Code  Function or Mode
                   ----  --------------------------------
                     P   Redefine Scratch Pad
                     D   Define List or Sequence
                     L   Operate on Lists
                     S   Operate on Sequences
                     C   Operate on Collations
                     A   Compare Ad Hoc
                     M   Direct Command Mode
                     O   Close Printer
                     ?   Help
                     .   Exit
                   ----  --------------------------------
             Code ..


Pad 164     166   Lists          Seqs          Colls          Reps
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  Exit                                                   Canc
```

The list definition has returned a list of 354 entries, and you will note that the list field at the foot of the screen now contains the number 3. This indicates that list number 3 is now defined and temporarily stored.

### Operate on Lists

Enter function code L in the Compare Utility Main Menu to manipulate the defined lists.

```
15:24:04           ***** PREDICT APPLICATION CONTROL *****          2000-02-29
User DBA              - COMPARE Utility List Headers  -        Library SYSPAC

List   1 Common 00164 00011     Library KSYSPAA
                                      Type NP   L
List   2 Common 00164 00010     Library SYSPAA
                                      Type NP   L
List   3 PAC    00164 00050 Compartment SYSPAC_214                       00017
                                      Type N    S


List   5 PAA    00164 00052    Location GSYSPAA  00164 00011
                                      Type N    L


List   7 Common 00164 00040
                                      Type D



Pad 164     135   Lists 123 5 7  Seqs 123456   Colls 1 3 5    Reps
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  Canc                                                   Canc
```

The Compare Utility List Headers screen shows list headers with some identification data. The list number, the kind value (Common, PAC or PAA), the type value (Natural, error messages, DDMs or foreign), as well as the library or application name, DBnr and Fnr is also part of the header record. This allows you to identify the list you have previously defined at a glance.

In order to compare list 1 with list 2, enter 1 in the input field underneath the first list and 2 underneath the second list to be compared. If you would prefer to compare list 2 with list 1, simply swap the numbers.

In the input field underneath each List line you can enter the following:

| | |
|---|---|
| 1 | or the first list to be compared |
| 2 | or the second list to be compared |
| D | To display the list |
| P | Print the list |
| W | Write the list to Natural workfile 1 |
| T | Save the list to a Natural text module |
| ? | for Help |

The following screen shows an example of the Display list function D. In our example we displayed list 1, which was a defined Common Natural list from DBnr=164, Fnr=11 and from library KSYSPAA for loadable program type objects.

```
15:46:39               ***** PAC COMPARE Utility  *****              2000-02-29
List   1 Common 00164 00011     Library KSYSPAA
                                      Type NP   L
  0001 ADJUST                          NP   L 2000-11-22 07:46:08 00001 00001
  0002 ADMIN                           NP   L 2000-01-04 11:15:35 00002 00002
  0003 CNVPAA22                        NP   L 2000-01-04 11:15:36 00002 00002
  0004 CNVPFTA                         NP   L 2000-01-04 11:15:36 00002 00002
  0005 CNVPLCR                         NP   L 2000-11-22 07:46:09 00001 00001
  0006 CNVPXREF                        NP   L 2000-11-22 07:46:10 00001 00001
  0007 CONVERT                         NP   L 2000-01-04 11:15:36 00002 00002
  0008 INTPAA                          NP   L 2000-11-22 07:46:11 00001 00001
  0009 MENU                            NP   L 2000-11-22 07:46:12 00001 00001
  000A PAAPIBEG                        NP   L 2000-12-01 15:38:34 00001 00001
  000B PA2PAACR                        NP   L 2000-02-08 09:55:06 00003 00000
  000C PA2PAAPL                        NP   L 2000-11-22 07:46:13 00001 00001
  000D PA2PADJ                         NP   L 2000-11-22 07:46:14 00001 00001
  000E PA2PAERR                        NP   L 2000-01-18 09:02:58 00002 00002
  000F PA2PAJOB                        NP   L 2000-11-22 07:46:15 00001 00001
  0010 PA2PALOC                        NP   L 2000-11-22 07:46:15 00001 00001
  0011 PA2PAMOD                        NP   L 2000-11-22 07:46:15 00001 00001
  0012 PA2PAMSC                        NP   L 2000-11-22 07:46:16 00001 00001
  0013 PA2PAOBJ                        NP   L 2000-11-22 07:46:16 00001 00001
Press ENTER to continue, enter '.' to stop:
```

The displayed list shows details including object name, object type, date and time and PAC and PAA version.

The following screen shows an example of the results of a comparison between list 1 and list 2. The results show by means of single character indicators the differences between the contents of list 1 versus list 2.

The second line displays the elements Li - List; 1M - list 1, common; NP - object group Natural Program; L - loadable; :2M - list 2, common; NP - object group Natural Program; L - loadable.

The third line displays the database ID, the file number and the library for both lists.

```
15:49:44               ***** PAC COMPARE Utility  *****              2000-02-29
Li  1M                       NP   L      :2M                       NP   L
    00164 00011 KSYSPAA                  :00164 00010 SYSPAA
  ! ADJUST   P 2000-11-22 07:46     00001 ADJUST   P 1999-05-28 22:15     00001
  ! ADMIN    P 2000-01-04 11:15     00002 ADMIN    P 1999-12-10 11:28     00001
  ! CNVPAA22 P 2000-01-04 11:15     00002 CNVPAA22 P 1999-07-03 23:27     00001
  ! CNVPFTA  P 2000-01-04 11:15     00002 CNVPFTA  P 1999-12-08 11:17     00001
  ! CNVPLCR  P 2000-11-22 07:46     00001 CNVPLCR  P 1999-11-23 13:18     00001
  ! CNVPXREF P 2000-11-22 07:46     00001 CNVPXREF P 1999-05-28 22:15     00001
  ! CONVERT  P 2000-01-04 11:15     00002 CONVERT  P 2000-01-19 11:48     00004
  <                                      CONVRT13 P 1999-06-21 13:24     00001
  <                                      CONVRT21 P 1999-06-21 13:24     00001
  > INTPAA   P 2000-11-22 07:46     00001
  ! MENU     P 2000-11-22 07:46     00001 MENU     P 1999-12-10 11:28     00001
  <                                      PAADC001 P 1999-01-16 12:59     00000
  <                                      PAADC012 P 1998-10-01 16:27     00000
  <                                      PAADC015 P 1998-09-30 18:44     00000
  > PAAPIBEG P 2000-12-01 15:38     00001
  ! PA2PAACR P 2000-02-08 09:55     00003 PA2PAACR P 2000-04-02 01:31     00001
  ! PA2PAAPL P 2000-11-22 07:46     00001 PA2PAAPL P 1999-12-10 11:28     00001
  ! PA2PADJ  P 2000-11-22 07:46     00001 PA2PADJ  P 2000-04-02 00:48     00001
  ! PA2PAERR P 2000-01-18 09:02     00002 PA2PAERR P 2000-01-19 11:48     00002
Press ENTER to continue, enter '.' to stop:
```

The (!) indicates that the object in list 1 has a timestamp different from that of its counterpart in list 2.

The (<) indicates that the object is present in list 2 but not in list 1.

The (>) indicates that the object is present in list 1 but not in list 2.

The (=) indicates that the timestamp of the objects coincide.

### Operate on Sequences

Enter function code S in the Compare Utility Main Menu to manipulate defined sequences.

An example of defined sequences is shown below:

```
15:51:59           ***** PREDICT APPLICATION CONTROL *****          2000-02-29
User DBA            - COMPARE Utility Sequence Headers  -      Library SYSPAC

Object 1 Common 00164 00011     Library KSYSPAA
     ADJUST                           Type NP   L
Object 2 Common 00164 00010     Library SYSPAA
     ADJUST                           Type NP   L
Object 3 PAC    00164 00050 Compartment SYSPAA_214                        00018
     CONVERT                          Type NP   S Version 00004
Object 4 PAC    00164 00050 Compartment SYSPAA_221                        00028
     CONVERT                          Type NP   S Version 00002
Object 5 PAC    00164 00050 Compartment SYSPAA_214                        00018
     ADJUST                           Type NP   S Version 00001
Object 6 PAC    00164 00050 Compartment SYSPAA_221                        00028
     ADJUST                           Type NP   S Version 00001





Pad 164     135   Lists 123 5 7  Seqs 123456   Colls 1 3 5 7  Reps
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  Canc                                                   Canc
```

The panel shows sequence headers with some identification data. The sequence number, the Kind value (Common, PAC or PAA), the Type value (Natural, error messages, DDMs or foreign), as well as the library or application name, object name, DBnr and Fnr is also part of the header. This allows you to identify the sequence you have previously defined at a glance.

In order to compare sequence 1 with sequence 3, enter 1 in the input field underneath the first sequence and 2 underneath the sequence 3. If you would prefer to compare list 3 with list 1, simply swap the numbers.

In the input field underneath each List line you can enter the following:

1 for the first sequence to be compared

2 for the second sequence to be compared

D Display the sequence

P Print the sequence

W Write the sequence to Natural workfile 1

E Display the directory information for the sequence

Q Print directory information for the sequence

X Write directory information to Natural workfile 1

? for Help

### Operate on Collations

Enter function code C in the Compare Utility Main Menu to manipulate collations.

An example of created collations is shown below:

```
15:52:58           ***** PREDICT APPLICATION CONTROL *****          2000-02-29
User DBA            - COMPARE Utility Collation Headers -      Library SYSPAC

Li  1M                       NP   L      :2M                       NP   L
    00164 00011 KSYSPAA                  :00164 00010 SYSPAA


Ob  3C CONVERT               NP   S 00004:4C CONVERT               NP   S 00002
    00164 00050 SYSPAA_214               :00164 00050 SYSPAA_221


Ob  5C ADJUST                NP   S 00001:6C ADJUST                NP   S 00001
    00164 00050 SYSPAA_214               :00164 00050 SYSPAA_221


Li  1M                       NP   L      :2M                       NP   L
    00164 00011 KSYSPAA                  :00164 00010 SYSPAA


The headers of collations with both arguments extant are highlighted.
Pad 164     135   Lists 123 5 7  Seqs 123456   Colls 1 3 5 7  Reps
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  Canc                                                   Canc
```

Collations are a result of the comparison of lists and sequences. When list 1 is compared with list 2 then the result of this is written to a collation, again using a number from 1 to 8.

These collation numbers can be seen in the Colls field at the bottom of the screen at all times. Once the collations have been created then it is possible to again manipulate the results of these stored collations.

The panel shows collation headers with some identification data. The source of the collation, the kind value (Common, PAC or PAA), the type value (Natural, error messages, DDMs or foreign), as well as the library or application name, Object name, DBnr and Fnr is also part of the header. This allows you to identify the collation you have previously created at a glance.

In the input field underneath each entry you can enter the following commands:

D Display the sequence

P Print the sequence

W Write the sequence to Natural workfile 1

E Display the directory information for the sequence
Q Print directory information for the sequence
X Write directory information to Natural workfile 1
R Create a report from the collation information
T Save the collation details to a Natural Text module
B Print bulk collations from collation
C Write bulk collations to Natural workfile 1
? for Help

### Compare Adhoc

The Compare adhoc function is very similar in its functionality to the Define List/Sequence function, except that with the adhoc function it is possible to define two lists or sequences from within the same panel. Enter function code A in the Compare Utility Main Menu.

The following screen illustrates the initial entry panel:

```
15:53:39           ***** PREDICT APPLICATION CONTROL *****          2000-02-29
User DBA            - COMPARE Utility Entity Definition -        Library SYSPAC

Entity (L,S)... _                                     Object Group (N,E,D,F,M) _
Number (1..8)   _                                   Kind (coMmon,paC,paA,Event) ↩
_









Number (1..8)   _                                   Kind (coMmon,paC,paA,Event) ↩
_









Pad 164     135   Lists 123 5 7  Seqs 123456   Colls 1 3 5 7  Reps
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  Retr        OK                                        Canc
```

The parameters of Entity, Number, Object Group and Kind are the same as previously defined. Once the above parameters have been filled, as the screen below shows, then the PF5 key becomes usable, as a confirmation key. Once this key is used then the completed screen's values are accepted as input and the utility then does an automatic comparison of the two defined entities and asks for a collation number to write away the results.

The following screen shows this:

```
15:53:39           ***** PREDICT APPLICATION CONTROL *****          2000-02-29
User DBA            - COMPARE Utility Entity Definition -      Library SYSPAC

Entity (L,S)... L                                     Object Group (N,E,D,F,M) N
Definition of list 6     (PAC)
Database number __164  File number ___50          Src/Loadable L
Application.... SYSPAA_214_______________________
Status......... ________________________________
Event.......... ________________________________
Object......... ________________________________ Object types _______________
PAC version.... _____                             Current on.. YYYY-MM-DD HH:II

Definition of list 4     (Common)
Database number 164__  File number 11___          Src/Loadable l
Library........ gsyspaa_                              +-------------------+
Object......... ________________________________ Ob  !  Placing of Result!
Time-fm........ YYYY-MM-DD HH:II                  Ti  ! collation 2       !
User........... ________                              +-------------------+


Pad 164     135   Lists 123 5 7  Seqs 123456   Colls 1 3 5 7  Reps
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  Retr        OK                                        Canc
```

The results of the above comparison are then displayed in collation 2 immediately below:

```
16:02:55                ***** PAC COMPARE Utility  *****              2000-02-29
Li  6C                        N    L 00000:4M                       N    L
    00164 00050 SYSPAA_214               :00164 00011 GSYSPAA
  = ADJUST   P 1999-05-28 22:15     00001 ADJUST   P 1999-05-28 22:15     00001
  = ADMIN    P 1999-12-10 11:28     00001 ADMIN    P 1999-12-10 11:28     00001
  > APIAADPD A 2000-01-29 10:21     00001
  > APIAADPS A 1999-05-28 22:13     00001
  > APIAAJOD A 1999-05-28 22:13     00001
  > APIAAJOS A 1999-05-28 22:13     00001
  > APIAALOD A 1999-05-28 22:13     00001
  > APIAALOS A 1999-05-28 22:13     00001
  > APIAAOBD A 1999-05-28 22:13     00001
  > APIAAOBS A 1999-05-28 22:13     00001
  > APIAPARM A 1999-05-28 22:13     00001
  = APINADPD N 2000-01-29 10:21     00001 APINADPD N 2000-01-29 10:21     00001
  = APINADPS N 1999-05-28 22:13     00001 APINADPS N 1999-05-28 22:13     00001
  = APINAJOD N 1999-05-28 22:13     00001 APINAJOD N 1999-05-28 22:13     00001
  = APINAJOS N 1999-05-28 22:13     00001 APINAJOS N 1999-05-28 22:13     00001
  = APINALOD N 1999-05-28 22:14     00001 APINALOD N 1999-05-28 22:14     00001
  = APINALOS N 1999-05-28 22:14     00001 APINALOS N 1999-05-28 22:14     00001
  = APINAOBD N 1999-05-28 22:14     00001 APINAOBD N 1999-05-28 22:14     00001
  = APINAOBS N 1999-05-28 22:14     00001 APINAOBS N 1999-05-28 22:14     00001
Press ENTER to continue, enter '.' to stop:
```

The use of the Compare adhoc is a quick method of achieving the desired results, from one central function. All of the defined entities are still temporarily stored as with the previous functions, and can then be manipulated as before.

### Extended Direct Command Mode

The Extended Compare utility may be executed using direct commands. To enter that mode, choose option (M) from the Extended Compare Utility main menu, at which point a blank screen is presented.

It may also be executed in batch at which point the direct command syntax is used. When using the COMPARE direct commands in batch the following Natural parameters must be used:

```
CF= OFF
ID= any character distinct from a ','
```

The former compare syntax is still valid with this version of the Compare Utility and the direct commands remain the same as documented later in this section. All former commands have to be preceded by the command COMPARE. For further information, see *The former COMPARE Command Syntax*.

A combination of both the former and new syntax is possible in a single session, however the new syntax must follow the former syntax because as soon as the new syntax is used then it is no longer possible to revert back to the former syntax. This is only valid for a single session.

The direct command syntax complies with the functionality that we have dealt with in the above menu functions. Define sequences, Define lists, Display lists, Display Collation, Display Object, Display Report and so on.

This section covers the following topics:



#### Explanation of Some of the Terms Used in the Direct Syntax:

| | |
|---|---|
| T1 or TYPE1 | this always has to be specified and indicates what type of objects is being dealt with.<br>Possible values:<br>F Foreign<br>N Natural<br>E Error Messages<br>D DDMs |
| T2 or TYPE2 | this value has to be specified for T1=Foreign and is optional for all other types of T1. Possible values: any user-defined Foreign object type. |
| T3 or TYPE3 | this value is optional depending on the value of T1. Possible values:<br>L Loadable<br>S Source |

#### Extended Compare Command Syntax

The Compare utility offers 14 new direct commands and "." for end. The direct commands consist of a combination of two letters, the first letter representing the compare entity, the second letter the function, for example PD: Pad Definition. The parameters valid for each command are described with each command type below.

P Pad

L List

S Sequence

R Report

D Define

T Natural Text

C Collate / Compare

P Print

In the following description these signs are used:

<text> further details about the parameter are given later in the section

| | |
|---|---|
| {text} | one of the parameter options listed must be entered |
| [text] | the element is optional |
| \| | or |
| italic | the variable value must be entered |
| * | you can use the item as often as you like or not at all |

**Define Commands**

The Define commands are used to create a sequence of objects for later comparisons. Choose one of the following commands using the syntax as described.

```
PD,<file definition>
LD,<number of list>,<kind>,{T1 | TYPE1},<type1>{,<selection criterion>}*
SD,<number of sequence>,<kind>,{T1 | TYPE1},<type1>{,<selection criterion>}*
RD,<number of  collation>{,<type of line of report>}*
```

Examples:

- Define a scratch pad file for use with the session on database 164, file 31.

```
pd,db,164,fnr,31
```

- Define list 1; create in it entries for all modules in (SYSPAC,164, 240) whose names begin with 'CF'.

```
ld,1,m,t1,n,t3,l,db,164,fnr,240,lib,syspac,obj,cf*
```

- Define list 2; create in it entries for all the latest sources of application SYSPAC_213 in ACF (164,50) whose names begin with 'MG'.

```
ld,2,c,t1,n,t3,s,db,164,fnr,50,appl,syspac_213,obj,mg*
```

- Define list 3; create in it entries for all the latest sources of application SYSPAC_214 in ACF (164,50) whose names begin with 'MG'.

```
ld,3,c,t1,n,t3,s,db,164,fnr,50,appl,syspac_214,obj,mg*
```

- Define sequence 1: Natural source named 'MGCNSVER' in application SYSPAC_213 in ACF (164,50), it shall be the highest numbered version if there are several.

```
sd,1,c,t1,n,t3,s,db,164,fnr,50,appl,syspac_213,obj,mgcnsver
```

- Define sequence 2: Natural source named 'MGCNSVER' in application SYSPAC_214 in ACF (164,50), it shall be the highest numbered version if there are several.

```
sd,2,c,t1,n,t3,s,db,164,fnr,50,appl,syspac_214,obj,mgcnsver
```

- Make a report of collation 2; omit matches, put into the report the non-matching lines of both the first and the second objects and the lines of either object that have no counterparts in the other object.

```
rd,2,!1,!2,>1,<2
```

| Parameter | Valid Values |
|---|---|
| File definition | {,{<DBNR phrase> \| <FNR phrase>}}*<br>For an explanation of the above elements, see next page. If the file definition is omitted, PCF is assumed. |
| Number of list | 1 to 8 |
| Number of sequence | 1 to 8 |
| Number of collation | 1 to 8 |
| Kind | M Common Objects known to Natural<br>C PAC objects under the control of PAC<br>A PAA objects under the control of PAA. |
| Type 1 | N Natural objects<br>E Error Messages<br>D Data Definition Modules<br>F Foreign Objects<br>See table of selection criteria below. |
| Type of line of report | =1, =2, !1, !2, <1, <2, >1, >2 |

| Parameter | Valid Values |
|---|---|
| Selection criterion | {APPL \| APPLICATION},<application name><br>AVNO,<PAA Version Number><br>{CURR \| CURRENT},<time><br>CVNO,<PAC version number><br><DBNR phrase><br>DSN,<PDS name><br>EBYTE,{<blank>,<LANGUAGE CODE>}<br>EVENT,<PAC event name><br><FNR phrase><br>JOB,<PAA job number><br>{LDB,LDBNR},<database number of a PAA location><br>LFNR,<file number of a PAA location><br><LIB phrase><br>{LT \| LTYPE \| VOL},<volume name or BS2000 LMS type><br>NODE,<node number><br>{OBJ \| OBJECT},<range of object names><br>STATE,<state of a PAA object><br>STATUS,<PAC status name><br>TIMEFM,<time><br>TIMETO,<time><br>{T2 \| TYPE2},{<Natural types> \| <foreign type>}<br>{T3 \| TYPE3},{L \| S}<br>USER,<user ID><br>For an explanation some of these elements, see below. |

Explanation of some of the above elements:

| Phrase | Syntax / Values |
|---|---|
| <DBNR phrase> | {DB \| DBNR},<database number> |
| <FNR phrase> | FNR,<file number> |
| <state of PAA object> | B (backed out), C (current), H (historical), R (removed) or S (scheduled). |
| <Natural types> | {<Natural type>}* |
| <time> | must fit mask: YYYYMMDDHHII. |

Comments:

- A DBNR or FNR phrase refers to an FPAA if <kind> is 'A', to an ACF if <kind> is 'C', to a Natural system file or an FDIC if <kind> is 'M'.
- Error messages should be specified with sequences of four digits.
- An EBYTE phrase is never a must. If none is present and error messages are being selected then they are short error messages.

- An EVENT phrase refers to an event that has brought objects into CONTROL if there is no STATUS phrase in the command or <PAC status name> is 'CONTROL'; otherwise it refers to an event that has taken objects to the deployment.
- A CURR phrase refers to a PAA object's currency in the location or to a PAC object's assumed currency in a deployment. A CURR phrase does not affect the selection of PAC objects if no status other than CONTROL is specified.

The following table shows which selection criteria are mandatory or optional when using the combination of the parameters <kind> and <type1>.
* optional
+ mandatory
B mandatory for BS2000, optional for all other operating systems.

| | AD | AE | AF | AN | CD | CE | CF | CN | MD | ME | MN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| APPL | * | * | * | * | + | + | + | + | | | |
| AVNO | * | * | * | * | | | | | | | |
| CURR | * | * | * | * | * | * | * | * | | | |
| CUNO | | | | | * | * | * | * | | | |
| DBNR | + | + | + | + | + | + | + | + | + | + | + |
| DSN | | | + | | | | | | | | |
| EBYTE | | * | | | | * | | | | * | |
| EVENT | | | | | * | * | * | * | | | |
| FNR | + | + | + | + | + | + | + | + | + | + | + |
| JOB | * | * | * | * | | | | | | | |
| LDBNR | + | + | | + | | | | | | | |
| LFNR | + | + | | + | | | | | | | |
| LIB | | + | | + | | | | | | + | + |
| LT | | | B | | | | | | | | |
| NODE | | | + | | | | | | | | |
| OBJ | * | * | * | * | * | * | * | * | * | * | * |
| STATE | * | * | * | * | | | | | | | |
| STATUS | * | * | * | * | * | * | * | * | | | |
| TIMEFM | | | | | | | | | | * | * |
| TIMETO | | | | | | | | | | * | * |
| TYPE2 | | | + | * | | | + | * | | | * |
| TYPE3 | | | + | + | | | + | + | | | + |
| USER | | | | | | | | | | | * |

*Collate / Compare Commands*

The Collate / Compare commands are used to perform the actual comparison of the previously defined sequences. The result of the comparison is written into a collation. Choose one of the following commands using the syntax as described.

```
LC,<number of list>,<number of list>,<number of collation>
SC,<number of sequence>,<number of sequence>,<number of collation>
 {,<comparison directive>}*
```

Examples:

- Collate lists 2 and 3; put the result in collation 1.

```
lc,2,3,1
```

- Compare sequences 1 and 2; put the result in collation 2.

```
sc,1,2,2
```

**Note:** A PAA object can be compared only if the object is backed up, current or scheduled.

| Parameter | Valid Values |
|---|---|
| Number of list | 1 to 8 |
| Number of sequence | 1 to 8 |
| Number of collation | 1 to 8 |
| Comparison directive | DIR \| DIRECTORY \| DISREGC \| DISREGCL<br>DIR(ECTORY): work on directory lines too<br>DISREGC: disregard tail comments<br>DISREGCL: disregard comment lines. |

**Print Commands**

The Print commands are used to manipulate the results obtained from a previously performed comparison. Choose one of the following commands using the syntax as described.

```
LP,<number of list>{,<print destination>}*
SP,<number of sequence>{,<print destination>}*
DP,<number of sequence>{,<print destination>}*
CP,<number of collation>{,{<print destination> | DISCR}}*
BP,<number of collation>{,{<print destination> | DISCR | <comparison directive>}}*
RP,<number of collation>{,<print destination>}*
```

Comments:

- RP will work only for a collation for which a report has been defined.

A PAA object or its directory can be printed only if the object is backed up, current or scheduled.

- DISCR in a CP command: the "matches" need not be printed;
  DISCR in a BP command: the "matching" couples of objects need not be compared (BP will anyway print only the "discrepancies" of each comparison of sequences it does.).

Examples:

- Display list 1.

```
lp,1,disp
```

- Display list 2.

```
lp,2,disp
```

- Display list 3.

```
lp,3,disp
```

- Display collation 1.

```
cp,1,disp
```

- Display sequence 1.

```
sp,1,disp
```

- Display sequence 2.

```
sp,2,disp
```

- Display collation 2.

```
cp,2,disp
```

- Display collation 2 showing only discrepancies.

```
cp,2,disp,discr
```

- Display report 2.

```
rp,2,disp
```

- Display the directory information of sequence 1.

```
dp,1,disp
```

- Display sequence 3.

```
sp,3,disp
```

| Parameter | Valid Values |
|---|---|
| Number of list | 1 to 8 |
| Number of sequence | 1 to 8 |
| Number of collation | 1 to 8 |
| Comparison directive | DIR \| DIRECTORY \| DISREGC \| DISREGCL |
| Print destination | DISP \| DISPLAY \| PRINT \| WORK \| WORKFILE \| PRINTC \| CPRINT \| CPRINTC |

Printing from Compare is available in the following modes that are reflected by the print destinations PRINTC CPRINT AND CPRINTC:

print-close
close-print
close-print-close

If no printer has been defined in the current COMPARE session, and the output has already been accumulated, it will noc close printer. The mode print-close will additionally close the printer. Close-print and close-print-close will begin by unconditionally close printer

To use printing from Compare, the settings of the NTPRINT macro and the PRINT parameter must ensure that:

printer 1 is available (in all cases)
CLOSE=CMD (if the closing is to work)
OPEN=ACC

**Natural Text Commands**

The Text commands are used to write the results of a comparison to a Natural text object in the Natural library you specified. Choose one of the following commands using the syntax as described.

```
LT,<number of list><NATURAL destination>
RT,<number of collation><NATURAL destination>
```

Examples:

- Write list 1 (see the very beginning of this scenario) to a Natural text named 'LIST1' in (ES-X-0, 164, 31).

```
lt,1,db,164,fnr,31,lib,es-x-0,obj,list1
```

- Write report 2 to a Natural text named 'LIST2' in (ES-X-0, 164, 31).

```
rt,2,db,164,fnr,31,lib,es-x-0,obj,list2
```

**Note:** RT will work only for a collation for which a report has been defined.

| Parameter | Valid Values |
|---|---|
| Number of list | 1 to 8 |
| Number of sequence | 1 to 8 |
| Number of collation | 1 to 8 |
| NATURAL destination | {,{<DBNR phrase> \| <FNR phrase> \| <LIB phrase> \| {OBJ \| OBJECT},<object name>}}* |

Explanation of some of the above elements:

| Phrase | Syntax / Values |
|---|---|
| <DBNR phrase> | {DB \| DBNR},<database number> |
| <FNR phrase> | FNR,<file number> |
| <LIB phrase> | {LIB \| LIBRARY),<library name> |

### The former COMPARE Command Syntax

This section covers the following topics:

- Overview
- WHERE-Clause
- WITH-Clause

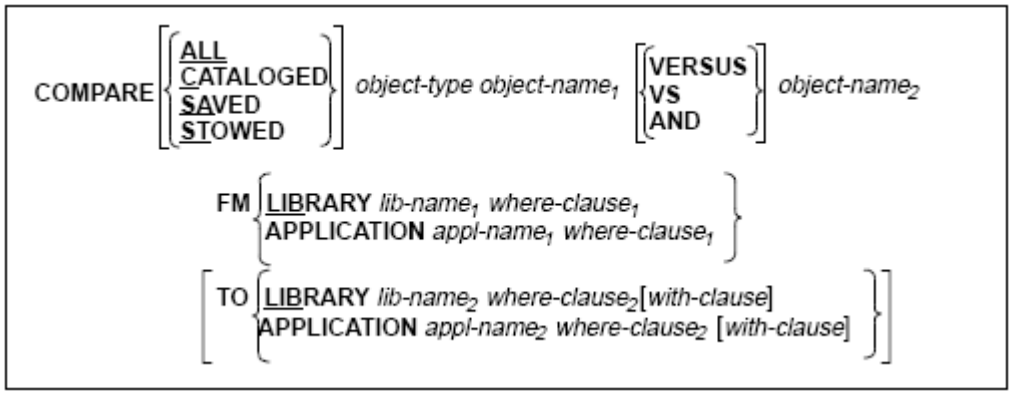
- Former COMPARE Command Examples

### Overview

This syntax is still valid for this version of PAC. However, there are the following important topics:

- the first command must always be COMPARE
- commas are required to separate syntax elements
- APPLICATIOn can no longer be abbreviated as APPL
- LIB or Library is a mandatory entry
- all commands need a QHERE clause indication DBnr's and FNR's.

The general COMPARE command syntax is as follows:




| Keywords | Explanation |
|---|---|
| ALL | Selects only saved components for proceessing |
| CAT | Selects only cataloged components for processing. (Any corresponding saved components will not be processed.) |
| SAVED | Selects all saved components for processing. (Any corresponding cataloged object will not be processed.) |
| STOWED | Selects all saved components for processing. (Any corresponding cataloged object will not be processed.) |
| IN/FM | Introduces the specification on the first list or object to be compared. |
| TO | Introduces the specification of the second list or object to be compared. |
| APPLICATION | Name of a PAC application that contains the object. |
| LIBRARY | Name of the library where the object exists. Mandatory. |

- COMPARE, *object-type*, and *object-name* must normally be the first three parameters of the command string.

- A period mark indicates the end of a command. If a "." is detected anywhere within a command string, all data after the "." will be ignored.
- The *library-name* must be specified immediately after the IN, FM, and TO keywords.
- The *application-name* must follow the APPLICATION keyword, which must be specified immediately after the IN, FM, and TO keywords.
- The where-clause must always follow the IN or FM keyword and *library-name* or *application-name*; however, the keywords and values within the clause may be specified in any order.
- The where-clause must always follow the TO keyword and *library-name* or *application-name*; however, the keywords and values within the clause may be specified in any order.
- The keywords and values of the with-clause may be specified in any order, and the with-clause may be placed anywhere within the direct command string, except in the first three positions.

### WHERE-Clause

The where-clause is mandatory. It specifies the loation of the library specified in the IN/FM (from) or TO clauses. The syntax is as follows:

```
WHERE [DBID dbid] [FNR file-nr] [NAME name] [CIPHER cipher]
[STATUS status-name] [VERSION version-number]
[PASSWORD password]
[SEC (dbid, fnr, psw, ciph)]
```

**Note:** The keywords and values within the clause may be specified in any order.

| Keywords | Explanation |
|---|---|
| WHERE | Indicates the start of a where-clause. |
| CIPHER | Cipher code of the file. (Ignored). |
| DBID | Database number of the specified library. |
| FNK | File number of the specified library |
| NAME | VSAM name (FCT) of the file (VSAM only). (Ignored). |
| PASSWORD | passowrd of the file. (Ignored). |
| SEC | DBnr, fnk, password, cipher of the Natural security file against which security validation must be performed. (Ignored). |
| STATUS | Name of the status. |
| VERSION | Version number of an / the object(s) to be compared. |

### WITH-Clause

The with-clause is optional. It specifies attributes of the objects to be selected. The syntax is as follows:

```
[WITH] [DIFFER] [EXPAND] [FMDATE date] [FMTIME time] [MON]
[SHORT] [TID terminal-id] [TODATE date] [TOTIME time] [TYPE type]
[USER user-id]
```

**Note:** The keywords and values of the with-clause may be specified in any order, ant the with-clause may be placed in any location within the direct command string, except in the first three positions.

| Keywords | Explanation |
|---|---|
| WITH | (Optional) Indicates the start of a with-clause. |
| DIFFER | Selects for processing all programming objects with different saved and cataloged dates. (Ignored). |
| EXPAND | Expands the compare report to show a comparison of both directories and source code; a short compare list is not displayed. The source comparison is written to CMPRT01. |
| FMDATE | Limits the selection by the saved/cataloge date. |
| FMTIME | Limits the selection by the save/catalog time. |
| MON | Activates online trace fycility (not valid in batch). (Ignored) |
| SHORT | Selects tor processing all programming objects found in the origin library. If SHORT is not specified, all objects in both libraries will be processed. (Ignored). |
| TID | Identifies the terminal ID where the saved/cataloged objects must have been processed to be selected. |
| TODATE | Identifies the save/catalog date after which no objects are selected. |
| TOTIME | Identifies the save/catalog time after which no objects are selected. |
| TYPE | Identifies the type(s) of objects to be selected. |
| USER | Identifies the ID of the user who saved/cataloged the objects to be selected for processing. |

### Former COMPARE Command Examples

To compare the saved components of a particular object in two different Natural libraries.:

```
COMPARE
COMPARE,ALL,,TRNSTOWP,FM,LIB,ASYSPAC,WHERE,DBID,164,FNR,11,%
TO,LIB,PAC14CSC,WHERE,DBID,164,FNR,11
.
```

To compare the cataloged component of a particular object in two different Natural libraries:

```
COMPARE
COMPARE,CATALOGED,,TRNSTOWP,FM,LIB,ASYSPAC,WHERE,DBID,164,FNR,11,%
TO,LIB,PAC14CSC,WHERE,DBID,164,FNR,11
.
```

To compare the saved component of a particular object in two different Natural libraries:

```
COMPARE
COMPARE,SAVED,,TRNSTOWP,FM,LIB,ASYSPAC,WHERE,DBID,164,FNR,11,%
TO,LIB,PAC14CSC,WHERE,DBID,164,FNR,11
.
```

To compare the saved component of a particular object in two different Natural libraries:

```
COMPARE
COMPARE,STOWED,,TRNSTOWP,FM,LIB,ASYSPAC,WHERE,DBID,164,FNR,11,%
TO,LIB,PAC14CSC,WHERE,DBID,164,FNR,11
.
```

To compare a range of components in two different Natural libraries:

```
COMPARE
COMPARE,ALL,,TRN*,FM,LIB,ASYSPAC,WHERE,DBID,164,FNR,11,%
TO,LIB,PAC14CSC,WHERE,DBID,164,FNR,11
.
```

To compare a component of an application in two different PAC statuses with different saved and cataloged dates:

```
COMPARE
COMPARE,ALL,,TRNSTOWP,IN,APPLICATION,SYSPAC_214,STATUS,CONTROL,%
WHERE,DBID,164,FNR,50,%
TO,APPLICATION,SYSPAC_214,WHERE,STATUS,PRODUCTION,%
WHERE,DBID,164,FNR,50,WITH,DIFF
.
```

To compare a range of components of an application in two different PAC statuses:

```
COMPARE
COMPARE,ALL,,TRN*,IN,APPLICATION,SYSPAC_214,STATUS,CONTROL,%
WHERE,DBID,164,FNR,50,%
TO,APPLICATION,SYSPAC_214,WHERE,STATUS,PRODUCTION,%
WHERE,DBID,164,FNR,50,WITH,DIFF
.
```

To compare a component of an application with two different versions:

```
COMPARE
COMPARE,ALL,,TRNSTOWP,IN,APPLICATION,SYSPAC_214,VERSION,2,%
WHERE,DBID,164,FNR,50,%
TO,APPLICATION,SYSPAC_214,WHERE,VERSION,3,%
WHERE,DBID,164,FNR,50
.
```

To compare a range of components of an application with two different versions:

```
COMPARE
COMPARE,ALL,,TRN*,IN,APPLICATION,SYSPAC_214,VERSION,2,%
WHERE,DBID,164,FNR,50,%
TO,APPLICATION,SYSPAC_214,WHERE,VERSION,3,%
WHERE,DBID,164,FNR,50
.
```

To compare a component of an application with two different versions, using the parameter *expand*:

```
COMPARE
COMPARE,ALL,,MENU,IN,APPLICATION,SYSPAC_214,VERSION,1,%
WHERE,DBID,164,FNR,50,%
TO,APPLICATION,SYSPAC_222,WHERE,VERSION,1,%
WHERE,DBID,164,FNR,50,WITH,EXPAND
.
```

# 4 Scan Utilities

Two scan utilities are available to the PAC user:

- Scan Migration Paths
  the scan utility for migration path verification and default update;
- Scan Natural Object Versions
  the scan utility for Natural object versions.

This chapter covers the following topics:

## Scan Migration Paths Utility - SCANPATH

This section covers the following topics:



The SCANPATH utility may be used to modify (update) the migration path defaults of several migration paths at a time. PAC administration functions also include a subset of the SCANPATH utility.

For example, the jobs used to process migration events can be updated for all migration paths that use a particular job.

The following options are available for modifying (updating) migration paths that satisfy your selection criteria:

- Modifying the default values for the migration paths that meet the specified selection criteria.
- Scan for migration paths that use a particular job and (optionally) replace that job, as well as other migration path defaults as follows:
  - Execution mode (batch/online)
  - Work file usage
  - Copy/move/include objects

### Accessing the SCANPATH Utility

To access the SCANPATH utility, logon to library SYSPAC, then enter the command SCANPATH at the NEXT prompt and press ENTER.

The Scan Migration Paths screen is displayed as shown in the following example:

```
 15:20:59          ***** PREDICT APPLICATION CONTROL *****           2000-04-16
 User SAGXX                - Scan Migration Paths -

 Function ............... _ (M-Modify, S-Scan/Replace, .- Exit)
 Confirmation Required .. Y

 Path Selection Criteria
   Application .......... *_______________________________
   From Status .......... *_______________________________  Type .. *
   To Status ............ *_______________________________  Type .. *

 Scan
   Job ................. _______________________________

 Replace
   Job ................. _______________________________
   Batch/Online ......... _
   Workfile Usage ....... _
   Copy/Move/Include .... _

Command ===>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit  Mod   Scan                                     Canc
```

The following table describes the attributes of the Scan Migration Paths screen that can be modified to satisfy your selection criteria:

| Attribute | Description |
|---|---|
| Function | Enter one of the following values:<br>"M" To modify the selected migration path defaults on the Modify Migration Path Defaults screen; or press PF4 (Mod).<br>"S" To access the Scan/Replace Migration Path screen in order to scan for a job used by all specified migration paths and replace it with another job; or press PF5 (Scan). The job can only be replaced if there is a replace value; otherwise, the user has the option to modify.<br>"(.)" To exit the function. |

| Attribute | Description |
|---|---|
| Confirmation Required | When S (Scan/Replace) is used in the Function field, the following values are valid:<br>"Y (Default)" Results in the user being prompted on the Scan/Replace Migration Path screen for the confirmation of each change before the change is made so that the user may update (by entering U), or ignore (by entering I) unwanted changes.<br>"N" To automatically update all migration paths that meet the selection criteria and display the resulting changes on a report screen. |
| Path Selection Criteria: | |
| Application | The name of the application for which the migration paths are defined. A range may be specified. |
| From Status | The origin status of the migration path(s). A range may be specified. |
| To Status | The destination status of the migration path(s). A range may be specified. |
| Type | The the status type. Valid values are as follows:<br>"A" Archive<br>"C" Control<br>"D" Development<br>"I" Incorporation<br>"M" Maintenance<br>"P" Production<br>"R" Retire<br>"T" Test |
| Scan: | |
| Job | The name of the current job used by the selected migration path(s). The name must be the complete name and not a string (as for an absolute scan). If left blank, all jobs are displayed. |
| Replace: | (Optional) Replace the following: |
| Job | The name of the job to replace the existing job for the selected migration path(s). If left blank, the existing job name is retained.<br><br>**Note:** The name of the job used to replace must exist; a check is performed to verify. A list is displayed on the Select Job window if you enter an asterisk (*) in this field. |
| Batch/Online | This field is an attribute of the migration path.<br>"B" The event is to be run in batch;<br>"O" The event is to be run online.<br>These values will override the existing values. If no replace value is specified, the existing value is retained. |
| Workfile Usage | "Y" A work file is to be used;<br>"N" A work file is not to be used.<br>These values will override the existing values. If no replace value is specified, the existing value is retained. |

| Attribute | Description |
|---|---|
| Copy/Move/ Include | This field is an attribute of the migration path.<br>C Objects are to be copied and still exist in the origin status;<br>M Objects are to be moved and not to exist in the origin status;<br>I Objects are to be included from a work file (from development or incorporation status types only).<br>These values will override the existing values. If no replace value is specified the existing value is retained. |

## Modifying Migration Path Defaults

This section describes the steps for modifying migration path defaults.

### Step 1. Access and Modify the Modify Migration Path Defaults Screen

1. On the Scan Migration Paths screen, enter M in the Function field and your selection criteria, and press ENTER; or press PF4 (Mod).

   The Modify Migration Path Defaults screen is displayed as shown in the following example:

```
15:21:23          ***** PREDICT APPLICATION CONTROL *****          2000-04-16
User SAGXX           - Modify Migration Path Defaults -
MODIFY Path Application ORD-EXAM FM Status * Type * TO Status * Type *
                                                                  Bat WKF Copy
Origin Status    Dest Status     Job                              Onl Use Move
---------------- --------------- -------------------------------- --- --- ----
ARCHIVE          CONTROL         ________________________________  B   Y   C
ARCHIVE          RETIRE          ________________________________  B   N   C
CONTROL          PRODUCTION      ________________________________  O   N   C
CONTROL          TEST            ________________________________  B   Y   C
CONTROL          MAINTENANCE     ________________________________  B   Y   C
CONTROL          ARCHIVE         ________________________________  B   Y   C
CONTROL          DEVELOPMENT     NATURAL_MIGRATE_________________  B   Y   C
CONTROL          CONTROL         ________________________________  B   N   C
CONTROL          RETIRE          ________________________________  B   N   C
DEVELOPMENT      TEST            ARCHIVE-PURGE-JCL_______________  B   N   C
DEVELOPMENT      USER_TEST       PREDICT_MIGRATE_IN______________  B   N   C
DEVELOPMENT      SYSTEMS_TEST    NATURAL_MIGRATE_________________  O   N   C
INCORPORATION    TEST            ________________________________  O   N   C
INCORPORATION    CONTROL         ________________________________  O   N   C


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit        Updat --           +           <     >    Canc
```

   This screen displays the direct command or request being processed under the User field, followed by the name of the application.

The Origin Status and Dest Status columns list all of the defined migration paths for the application. The status names are highlighted if they have been truncated.

The attributes that you are allowed to modify on the Modify Migration Path Defaults screen are highlighted on terminals which have the highlighting facility.

The following information may be modified on this screen:

| Column | Description |
|---|---|
| Job | The name of the current job. |
| Bat/Onl | Whether the event is run in batch (B) or online (O). If O (online) is specified, the value in the WKF Use field must be N. |
| WKF Use | Whether a work file is used (Y) or not (N). |
| Copy/Move | Whether the Copy option (C), the Move option (M), or the Include (I) option is specified. If the destination status is a Retire status type, M (move) must be specified. |

The cursor moves to each value that requires modification until all values on the screen have been correctly modified with a valid value based on the origin and destination status types.

2. Make the desired modifications.
3. Press ENTER to verify your modifications.

**Note:** If you enter a new job name, a verification check is made to ensure the job exists; if not, an error is issued. If you enter an asterisk (*) in the Job name field, a selection list is displayed.

#### Step 2. Confirm the Modifications

Press PF5 (Upd) to confirm your modifications.

An information message is displayed indicated the total number of migration paths that have been updated.

To cancel your modifications and display the Scan Migration Paths screen, press PF3. To cancel your modifications and display the next screen (where available), press PF8.

#### Step 3. Display Additional Information

Display additional information about the selected migration paths on the Modify Migration Path Defaults screen as follows.

1. PF10 (<) displays the application name, the job name, and the type of migration path in the Path column. On this screen, the job names are modifiable.

```
15:21:42          ***** PREDICT APPLICATION CONTROL *****          2000-04-16
User SAGXX           - Modify Migration Path Defaults -
MODIFY Path Application ORD-EXAM FM Status * Type * TO Status * Type *

Application                      Job                              Path
-------------------------------- -------------------------------- ------------
ORD-EXAM                         ________________________________ Archv->Cntrl
ORD-EXAM                         ________________________________ Archv->Retre
ORD-EXAM                         ________________________________ Cntrl->Prod
ORD-EXAM                         ________________________________ Cntrl->Test
ORD-EXAM                         ________________________________ Cntrl->Maint
ORD-EXAM                         ________________________________ Cntrl->Archv
ORD-EXAM                         NATURAL_MIGRATE_________________ Cntrl->Devlp
ORD-EXAM                         ________________________________ Cntrl->Cntrl
ORD-EXAM                         ________________________________ Cntrl->Retre
ORD-EXAM                         ARCHIVE-PURGE-JCL_______________ Devlp->Test
ORD-EXAM                         PREDICT_MIGRATE_IN______________ Devlp->Test
ORD-EXAM                         NATURAL_MIGRATE_________________ Devlp->Test
ORD-EXAM                         ________________________________ Incrp->Test
ORD-EXAM                         ________________________________ Incrp->Cntrl


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit        Updat --           +           <     >    Canc
```

2. PF11 (>) displays the origin status, the destination status, and the type of migration path in the Path column. These attributes are not modifiable.

```
15:21:53          ***** PREDICT APPLICATION CONTROL *****          2000-04-16
User SAGXX           - Modify Migration Path Defaults-
MODIFY Path Application ORD-EXAM FM Status * Type * TO Status * Type *

Origin Status                    Destination Status               Path
-------------------------------- -------------------------------- ------------
ARCHIVE                          CONTROL                          Archv->Cntrl
ARCHIVE                          CANCEL                           Archv->Retre
CONTROL                          PRODUCTION                       Cntrl->Prod
CONTROL                          TEST                             Cntrl->Test
CONTROL                          MAINTENANCE                      Cntrl->Maint
CONTROL                          ARCHIVE                          Cntrl->Archv
CONTROL                          DEVELOPMENT                      Cntrl->Devlp
CONTROL                          CONTROL                          Cntrl->Cntrl
CONTROL                          RETIRE                           Cntrl->Retre
DEVELOPMENT                      TEST                             Devlp->Test
DEVELOPMENT                      USER_TEST                        Devlp->Test
DEVELOPMENT                      SYSTEMS_TEST                     Devlp->Test
INCORPORATION                    TEST                             Incrp->Test
INCORPORATION                    CONTROL                          Incrp->Cntrl


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit        Updat --           +           <     >    Canc
```

## Scanning/Replacing Migration Path Defaults

The procedure for scanning/replacing migration path defaults differs depending on whether you need to confirm updates before they are implemented.

### Confirmation Required

When the Confirmation Required attribute on the Scan Migration Paths screen is set to Y (Default), the user is prompted for confirmation of each change before the change is made so that unwanted changes can be ignored. The following procedure is used if confirmation of changes is required:

**Step 1. Access the Scan/Replace Migration Paths Screen**

From the Scan Migration Paths screen, either press PF5 (Scan) or enter the following and press ENTER:

1. The relevant selection criteria;
2. S (Scan/Replace) in the Function field; and
3. Y in the Confirmation Required field.

The Scan/Replace Migration Path screen appears as shown in the following example:

```
 15:22:15          ***** PREDICT APPLICATION CONTROL ****          2000-04-16
 User SAGXX            - Scan/Replace Migration Path -

 Application .......... ORD-EXAM
 Origin Status ........ ARCHIVE                          Type .. Archive
 Destination Status ... CONTROL                          Type .. Control


 Update Path .......... U (U-Update, I-Ignore)

 Current Values
   Job ................
   Batch/Online ....... B           Workfile Usage ... Y
   Copy/Move/Include .. C

 New Values
   Job ................ ________________________________
   Batch/Online ....... B           Workfile Usage ... Y
   Copy/Move/Include .. C

PAC7000: (I) Enter 'U' to update Path, 'I' to Ignore.
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit        Updat                                     Canc
```

The Scan/Replace Migration Path screen displays the first migration path that satisfies the selection criteria. The following table describes the fields on this screen that indicate the current defaults for the specified migration path. These fields cannot be modified.

| Attribute | Description |
|---|---|
| Non-modifiable: | |
| Application | The name of the application to which the migration path (origin/destination statuses) is defined. |
| Current Values | The current values for the following: |
| Job | The name of the existing job for the selected migration path(s). |
| Batch/Online | This field is an attribute of the migration path.<br>"B" The event is to be run in batch;<br>"O" The event is to be run online. |
| Workfile Usage | This field is an attribute of the migration path.<br>"Y" A work file is to be used;<br>"N" A work file is not to be used. |
| Copy/Move/ Include | This field is an attribute of the migration path.<br>"C" Objects are to be copied and still exist in the origin status;<br>"M" Objects are to be moved and not to exist in the origin status;<br>"I" Objects are to be included from a work file (from development or incorporation statuses only.) |

| Attribute | Description |
|---|---|
| Destination Status | The destination status of the selected migration path. |
| Origin Status | The origin status of the selected migration path. |
| Type | The status type of the specified origin and destination statuses. |

**Step 2. Update the Migration Path Defaults**

The remaining fields on the screen are input fields where you can enter values to update the migration path defaults, if required.

1. Review the current default information:
   - If you want to update the current migration path defaults, enter U in the Update Path field.
   - If you do not want to update this migration path, enter I to ignore, then press ENTER to display the next migration path.
2. If you entered U in the Update Path field, enter new values in the New Values fields as described in the following table:

| Attribute | Description |
|---|---|
| New Values | New values that you can enter for the following: |
| Job | The name of the job to replace the existing job for the selected migration path(s). If left blank, the existing job name is retained. |
| Batch/Online | B The event is to be run in batch; O The event is to be run online. These values will override the existing values. If no replace value is specified, the existing value is retained. (This field is an attribute of the migration path.) |
| Workfile Usage | Y A work file is to be used; N A work file is not to be used. These values will override the existing values. If no replace value is specified, the existing value is retained. (This field is an attribute of the migration path.) |
| Copy/Move/ Include | C Objects are to be copied and still exist in the origin status; M Objects are to be moved and not to exist in the origin status; I Objects are to be included from a work file. These values will override the existing values. If no replace value is specified, the existing value is retained. (This field is an attribute of the migration path.) |

3. Press ENTER or PF5 (Upd) to verify the modifications.

When the currently displayed migration path has been modified, the next migration path that meets the specified selection criteria is displayed. This process continues until all of the specified migration paths have been displayed/updated.

To cancel the process at any time before all of the migration paths have been reviewed and/or updated, press PF3 (Exit) or PF12 (Canc).

When all of the migration paths that meet the specified selection criteria have been reviewed and/or updated, the following message is displayed: "... Migration Path(s) has been modified." where "..." is the total number of migration paths.

#### Confirmation Not Required

Where the Confirmation Required attribute on the Scan Migration Paths screen is set to N, no confirmation is required and the changes are automatically made for the selected migration paths. The following procedure is used if confirmation of changes is not required:

1. From the Scan Migration Paths screen, enter the following and press ENTER:
   - The relevant selection criteria;
   - S (Scan/Replace) in the Function field; and
   - N over the Y default in the Confirmation Required field.

   The selected migration paths for the specified application(s) are updated and the following Scan/Replace Migration Paths screen is displayed:

```
15:21:53          ***** PREDICT APPLICATION CONTROL *****          2000-04-16
User SAGXX           - Scan/Replace Migration Paths


Application           Origin Status         Destination Status       Message
--------------------- --------------------- ---------------------    ---------
ORD-EXAM              CONTAROL              DEVELOPMENT              Updated
ORD-EXAM              DEVELOPMENT           USER_TEST                Updated
ORD-EXAM              INCORPORATION         CONTROL                  Updated
ORD-EXAM              SYSTEMS_TEST          PRODUCTION               Updated
ORD-EXAM              USER-TEST             SYSTEMS-TEST             Updated









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit        Upd   --           +           <     >    Canc
```

   This screen displays the migration paths (Origin/Destination statuses) for the application(s) and the message "Updated" in the Message column for each migration path updated. The information on this screen cannot be modified.

2. Display additional non-modifiable information about the migration paths as follows:
   - PF10 (<) displays the application name, the job name, and the type of migration path in the Path column.
   - PF11 (>) displays the origin status, the destination status, and the type of migration path in the Path column.

   Refer to Step 3.**Display Additional Information** for screen examples.

## Scan Natural Object Version Utility - SCANOBJ

This sectioncovers the following topics:



The SCANOBJ utility allows you to scan for any string of characters in the source code of versioned objects in an application defined to PAC. The utility can be run in batch or online.

For any one scan request, the SCANOBJ utility scans objects in one particular application only. The application must be valid and defined to the current PAC.

The object(s) scanned may be any of the valid Natural object types; for example, program, subroutine, map, data area, help routine.

Each line in a specific versioned object (except for maps) that is found to satisfy the scan criteria is displayed as if it were in the normal editor for that object. These lines may be viewed, but not modified.

If the scan string is set to absolute, PAC looks for the existence of the string anywhere in the object source code. If the scan string is not set to absolute, only those instances of the requested string that are surrounded (delimited) by blanks are scanned.

The SCANOBJ utility allows you to scan an application for

- a single object;
- a range of objects beginning with the same name character string;
- all objects currently in a specific status;
- all objects with a specific type;
- all objects within a specific version range.

Scan by Status.
When When scanning objects in a particular status, the status must be valid and linked to the application specified. PAC ignores any version range settings; object type criteria are respected.

Scan by Version Number.
When scanning a range of objects with a range of version numbers or a specific version number, the status must be set to CONTROL. CONTROL is the default. Object type criteria are respected.

If the current (most recent) versions are to be scanned, the From Version and To Version numbers must be set to 0 (zero) and the status set to CONTROL.

### User Exit for SCANOBJ

The scan utility exit, User Exit PACEX035, is invoked with the SCANOBJ utility for every object to be scanned in PAC. PACEX035 allows the user to perform security checking for each request and then for each object at the time the SCANOBJ utility is invoked.

Refer to section **User Exits** for information about coding this user exit.

### SCANOBJ Processing Rules

The following rules apply when using SCANOBJ:

- If the source for an object is not in PAC (for example, the object has been archived, or the object source was not incorporated into PAC), a message to that effect is displayed and the object is ignored.
- Lines containing PASSWORD=, PASSW=, CIPHER=, or CIPH=, are ignored as a security precaution.

### Invoking the SCANOBJ Utility

When invoked, the SCANOBJ utility clears the object source area before scanning. Therefore, you must SAVE (STOW) any object source you may be modifying before you invoke any SCANOBJ function.

SCANOBJ may be invoked using keywords, or directly as a command in either batch or online mode. The SCANOBJ utility may be invoked in the following ways:

- Online, using the Scan Natural Object Version Utility screen;
- Online, using keywords;
- In batch, using keywords;
- In batch, using batch commands.

If the SCANOBJ and SCN* routines are in library SYSTEM, then the user need not log onto PAC; however, the PAC NTFILE 210 must be set.

### Online SCANOBJ Utility

1. To access the SCANOBJ utility online, logon to library SYSPAC, then enter the command SCANOBJ at the NEXT prompt.

   The SCAN Natural Object Version Utility screen is displayed as shown in the following example:

```
 15:20:59          ***** PREDICT APPLICATION CONTROL *****          2000-04-01
 User SAGXX        - SCAN NATURAL Object Version Utility -      Library SYSPAC

 Function ......... S
 Application ...... ________________________________

 Object
   Name ........... ___________
   Type ........... ___________
   Status ......... CONTROL_________
   Version ........ 1___ To .. 9999


 Scan Parameters
   String ......... ________________________________
   Absolute Scan .. N
   Trace Option ... N




Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit
```

2. Enter the selection criteria on this screen using the following table as a reference.

| Attribute | Information to be Entered |
|---|---|
| Function | Enter one of the following values:<br>"S" to scan the object(s) of an application.<br>"?" to view the help facility of SCANOBJ.<br>"." to exit the SCANOBJ utility. |
| Application | (Required) The PAC application to which the Natural object(s) belong. |

| Attribute | Information to be Entered |
|---|---|
| Object Name | Enter a single object name or range notation for some or all object names as follows:<br>* To scan all object names, leave the field blank or enter an asterisk.<br>...* To scan only object names beginning with particular characters, enter those characters followed by an asterisk.<br>...><br>...<<br>To scan only object names in one part of the alphabet, enter the beginning characters followed by a greater than (>) or less than (<) sign. The list of object names may be limited by entries for object type, status, and version. |
| Object Type | Enter one or more object type codes as follows: a combination of object types, or all object types. If a specific type is selected, it will limit the number of objects scanned. A combination of any of the following values is permitted:<br>* To scan all object types, leave the field blank or enter an asterisk.<br>n... To scan one or more object types, enter the code for each object type to be scanned in any order or combination (for example, GAL) where n is:<br>"A" Parameter data area<br>"C" Copy code<br>"G" Global data area<br>"H" Help routine<br>"K" Server<br>"L" Local data area<br>"M" Map<br>"N" Subprogram<br>"P" Program<br>"R" Report<br>"S" Subroutine<br>"T" Text<br>"Z" Recording<br>"3" Dialog<br>"4" Class<br>"5" Processor |
| Object Status | To limit the scan to objects which have been promoted to a particular PAC status, enter the status name. The default is CONTROL which contains all objects defined to PAC. |
| Object Version | To limit the object version levels that will be scanned, enter the From and, optionally, the To object version numbers. When specified, a To version number cannot be greater than a From version number unless To is 0 (zero). To scan only the most current versions, set both To and From to 0 (zero). The default (1-9999) scans all object versions. |
| Scan Parameters: | |
| String | (Required) The value or character string to be scanned for in each object. |
| Absolute Scan | If Y, PAC scans for the string anywhere in the object source. If N (the default), PAC scans for the string only when it is surrounded (delimited) by blanks. |
| Trace Option | To display a window listing the object type and version being processed, change the default N to Y. Do not turn on the trace option if the user's terminal does not support the SET CONTROL N facility. |

Keywords can be used to invoke the SCANOBJ utility either online or in batch. A list of valid keywords is described in the following table:

| Keywords | Description |
|---|---|
| FUNC | Function |
| TYPE | Object type |
| SVAL | String value |
| APPL | Application |
| STATUS | Status name |
| ABSOL | Absolute scan |
| OBJ | Object name |
| FMVER | From version number |
| TOVER | To version number |

**Example**

The following example of SCANOBJ invoked using keywords indicates that all subroutines are to be scanned with names that begin with PGM, in application ORDERS, in the Production status, for the string FIND:

```
SCANOBJ FUNC=S,APPL=ORDERS,OBJ=PGM*,TYPE=S,STATUS=PRODUCTION,
 SVAL=FIND
```

### SCANOBJ Utility in Batch

The SCANOBJ utility can be invoked in batch. The parameters are positional and must be specified in the following order:

```
SCANOBJ S, application, object name, object type, status, From version, To ↩
version, scan string, absolute indicator
```

**Example**

The following example of SCANOBJ invoked using positional parameters indicates that all subroutines are to be scanned with names that begin with RTN, in application ORDERS, in the Production status, for the string FIND:

```
SCANOBJ S,ORDERS,RTN*,S,PRODUCTION,,,FIND,Y
```

### Using SCANOBJ Results

When you press ENTER after entering your search criteria information (online or in batch, using screens or commands), the Natural object version information that satisfies your selection criteria is displayed.

**Example 1**

```
Applic .. ORD-EXAM                          Object ... APGMP2.0011 (Progrm)
String .. M                                 Command .. ____________________
       ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
 0010  DEFINE DATA LOCAL USING APGML2
 0030  WRITE '93-05-31  10:49:06         - VERSION 0001'  *program
 0040  INCLUDE APGMC1
 0050  INPUT USING MAP 'APGMM1'
 0060  CALLNAT 'APGMN1' AAA BBB
 0070  PERFORM APGM-RTN1
 0080  FETCH RETURN 'APGMP5'
 0090  WRITE '93-05-31  10:49:19         - VERSION 0001'  *PROGRAM
 0100  FETCH 'APGMP3'
```

The example screens display the following information:

- The name of the application to which the selected object belongs;
- The object name and type that meets the selection criteria;
- The specified scan string (in the above screen example, "M") in the String field;
- A Command line on which you may enter commands;
- The line numbers and the lines of source code of the object (in the above example, a Program) that contain the specified scan string.

**Note:** Currently, the SCAN editor limits the line length of the object source to 72.

#### SCANOBJ Commands

The following commands may be issued on the command line of the screens that satisfy your selection criteria:

| Command | Description |
|---|---|
| Blank + ENTER | Continue with the normal processing of the SCANOBJ utility for the next object that meets the specified selection criteria. |
| Q or PF3 | Exit the current SCANOBJ processing for the specified selection criteria and return to the menu. |
| I | Ignore the object currently being scanned and continue with the next object. This is useful if more than one page of output is being displayed. |

**Example 2**

The following screen example shows parameter data area APGMA1.0006 in the application ORD-EXAM containing the requested string "A" (String field):

```
Applic .. ORD-EXAM                          Object ...  APGMA1.0006 (Param)
String .. A                                 Command .. ____________________
 I T L Name                             F Leng  Index/Init/EM/Name/Comment
 - - - --------------- ALL ------------ - ---- ------------------------------
     1 AAA                              A   10 /* VERSION 0003 04.06.92
     1 BBB                              A    9 INIT<'TESTL2-03'>
```

**Example 3**

The window in the following screen example appears when an object that meets the specified selection criteria is encountered, but it has no corresponding source code; that is, it exists in PAC only as a cataloged object. The SCANOBJ utility ignores such an object.

```
Applic .. ORD-EXAM                          Object ... APGMA1.0006 (Param)
String .. A                                 Command .. ____________________
 I T L Name                             F Leng  Index/Init/EM/Name/Comment
 - - - --------------- ALL ------------ - ---- ------------------------------
     1 AAA                              A   10 /* VERSION 0006 ALIGN  002
     1 BBB                              A    1


                                     +--------------------------------+
                                     | No Saved object exists for ..  |
                                     |   APGMA1   (PARAM)  ver  0005  |
                                     | **  Object has been ignored ** |
                                     |                                |
                                     +--------------------------------+
```

# 5 User Exits

Each user has unique change management requirements that may vary from time to time, or with different situations or special circumstances.

PAC accommodates the user's needs by providing user exit facilities. Control can be passed to user-written Natural programs (user exit routines) to override the change management procedure defaults established by PAC.

The following sections describe PAC user exits in detail. Refer to the General Defaults Maintenance chapter in the PAC Administration documentation for information about displaying and modifying PAC user exits.

This chapter covers the following topics:

## User Exit Groups

PAC groups its user exits as follows:

| Group | Members | Description |
|---|---|---|
| **General** | 1, 2, 33, 40 | PAC initialization, termination; node security check |
| **Maintenance** | 4, 8, 13-17, 19 - 21, 25, 26, 30, 38 | Custom maintenance of PAC entities |
| **Migration** | 5-7, 9-12, 23, 27, 28, 31, 32 | Customized migration events |
| **Reporting** | 18, 22, 24 | Customized access to versioned object information |
| **Special Use** | 34, 36, 37 | Customize object purges in origin status with Move option; customize view generation for Predict migration, direct commands |
| **Scan Utility** | 35 | Exits for the scan utility |
| **Migration Utility** | MIGEX003 (3) | Exits for the migration utilities, file security |
| Not assigned: | 29, 33, 39, 41-48 | |

## Overview of Exits

This sectioncovers the following topics:

- General Group
- Maintenance Group
- Migration Group
- Reporting Group
- Special Use Group
- Scan Utility Group
- Migration Utility Group

- Common Data Area used by all Exits

## General Group

| Exit | Description |
|---|---|
| PACEX001 | PAC initialization: invoked during the initialization of PAC; allows the user to pass a 50-byte parameter area throughout the session. |
| PACEX002 | PAC termination: invoked during the termination of PAC; any required processing may be done in this routine. |
| PACEX033 | PAC Activity Authorisation Exit; used if PAC runs unter security. |
| PACEX040 | Security check for non-Natural datasets and nodes. When using Entire System Server to access a remote node, this exit allows you to specify a different user ID and password and the password for protected PDSs. |

## Maintenance Group

| Exit | Description |
|---|---|
| PACEX004 | PAC entity purge validation: invoked before the deletion of a PAC entity; allows the user to interrogate the entity to be deleted before processing by PAC. |
| PACEX008 | Application status library validation: invoked during the creation of application status links for a particular application; allows the user to validate the specific libraries assigned to each specific application status link. |
| PACEX013 | Migration event name/access validation: invoked whenever any migration event maintenance function is invoked; can be used to implement audits and security. |
| PACEX014 | Status name/access validation: invoked whenever any status maintenance function is invoked; can be used to implement audits and security. |
| PACEX015 | Application name/access validation: invoked whenever any application maintenance function is invoked; can be used to implement audits and security. |
| PACEX016 | File translation table name/access validation: invoked whenever any FTT maintenance function is invoked; can be used to implement audits and security. |
| PACEX017 | Application status link access validation: called whenever any migration paths function is invoked; can be used to implement audits and security. |
| PACEX019 | Maintenance request name/access validation: invoked during the processing of a maintenance request subfunction. Because information can be passed to or from the exit, maintenance request information can be verified or derived from an external problem-tracking system. |
| PACEX020 | Maintenance request processing validation: invoked for maintenance request processing whenever any maintenance request function is invoked; allows the user to implement audits and security. |
| PACEX021 | Maintenance request name verification: invoked to validate the maintenance request name attribute whenever a migration event is added or modified; may be used to implement audits and security. |

| Exit | Description |
|---|---|
| PACEX025 | Migration event names: invoked when function code A (Add) or C (Copy) is selected from the Migration Event Maintenance menu; allows the user to implement automated migration event name generation/ verification routines. |
| PACEX026 | Verification of object list selection for Archiving: invoked during the onset of the generation of an object list for an Archive unload event; allows the user to set up and/or verify the criteria for selecting the objects to be archived. |
| PACEX029 | Test Production Emigration: PACEX029 is called at the application of an FTT to a module being emigrated to a test or production location: once for each reference to a file in the module to which the FTT is being applied. |
| PACEX030 | Event authorization verification: invoked before and after authorization begins; may be used to implement any required audits or security. |
| PACEX038 | Event authorization verification: invoked before and after authorization begins; may be used to implement any required audits or security. Will replace PACEX030. |

## Migration Group

| Exit | Description |
|---|---|
| PACEX005 | Locking objects: invoked before locking an object for out-migrations only as well as when the 'from-status' is CONTROL. Non-zero response value will reject the object from locking and taking no further part in the migration. Is invoked once for every object that is to be locked. |
| PACEX006 | Compilation Error Verification: invoked after the compile processing step of the event is completed when any compilation errors are detected. Normally, PAC would force the event to abnormally terminate; however, this exit may be used to prevent the termination. |
| PACEX007 | Change control list validation: invoked during the check-out of objects to a development/ maintenance type status and check-in from a development/maintenance type status, although change control logs are created only for the maintenance status type. The user may accept or reject the request to process each object in the event object list. |
| PACEX009 | Verification of duplicate objects for migration. When PAC determines that the object being checked in from development or maintenance has the same date-time stamp as the most recent version in CONTROL, the object is treated as a duplicate. Normally, PAC creates a new version of all objects except copy code and text. This exit allows you to override PAC's decision to create a new version. |
| PACEX010 | Object list verification: invoked for each entry in the object list when PAC is validating the objects to be processed for a migration event. As each entry in the object list is passed to the exit, the PAC administrator has the option to disallow that entry. |
| PACEX011 | Job validation: called when a job maintenance subfunction is invoked; allows the user to implement tracking and customized security. |
| PACEX012 | Job submission verification: invoked during JCL submission; passes the first 10 lines of JCL so that the user can examine them and make any desired changes or substitutions prior to job submission. |
| PACEX023 | Object verification for autoexpand. During the Expand option processing of a migration event, this user exit is called before a particular object is expanded. PAC passes the object and the |

| Exit | Description |
|---|---|
| | Expand option to the exit to be verified. The user may change the Expand option for the particular object or suppress the expansion of the object. |
| PACEX027 | Verification of user substitution parameters for jobs: called whenever a batch job with user substitution parameters is submitted from PAC. Because it may be called before and after the substitution parameters are specified, the exit may be used to specify the default parameters before the job is presented to the user or for verification after the user has modified the job. |
| PACEX028 | Migration audit report message verification: invoked during migration event processing, usually when a message is being written to the audit report; may optionally be used to suppress the writing of an audit message to the audit report. |
| PACEX031 | Pass CATALL parameters to PAC: invoked prior to the CATALL; used for compiling objects in PAC during migrations from development or maintenance. |
| PACEX032 | Reject objects for object list: invoked for each object elected by PAC during object list selection from the object list editor; used to reject objects based on a moving nonzero value for the MSG-NO parameter (Natural Optimizer Compiler). |

## Reporting Group

| Exit | Description |
|---|---|
| PACEX002 | Object versions 'used by list' reporting. When the Expand option of a migration event is in effect during the processing of a migration event, PAC audits and passes for verification each subordinate object for the object being processed. This facilitates the expansion of objects defined across applications; that is, STEPLIB applications. |
| PACEX018 | Object version reporting: invoked whenever any versioned objects reporting function is invoked; can be used to implement audits and security. |
| PACEX024 | Reporting user exit to obtain the before and after images of a migration list. |

## Special Use Group

| Exit | Description |
|---|---|
| PACEX034 | Verification of PAC direct commands: allows you to process a string of characters before it is submitted to PAC as a direct command. |
| PACEX036 | Object origin deletion: When the Move option is in effect for a batch work file migration, this exit provides the ability to generate CMSYNIN cards to be used with a batch Natural utility (SYSMAIN, MIGUNLD) for purging objects in the origin status. |
| PACEX037 | Predict generation: an interface exit that may be used to suppress the generation of views during Predict migrations. Additionally, in the case of files linked to multiple databases, allows you to override the database ID for which the generation must be performed. PAC uses information from the generation defaults to determine which default database to use. This is only valid for files linked to multiple databases. |

### Scan Utility Group

| Exit | Description |
|---|---|
| PACEX035 | The scan utility exit, User Exit PACEX035, is invoked with the SCANOBJ utility for every object to be scanned in PAC. |

### Migration Utility Group

| Exit | Description |
|---|---|
| MIGEX003 | Security check for Natural or Predict files: invoked whenever PAC needs to access a Natural or Predict file to retrieve or store Natural or Predict objects; allows the user to establish the password and cipher code to be used when PAC accesses these files. |

### Common Data Area used by all Exits

PAC user exits are delivered in library SYSPACUS and serve as an example only on how the user exit is to be coded.

All PAC user exits use a standard Parameter Data Area PACEXPDA, and it is explained in the table below:

| Parameter | Format | Data Area |
|---|---|---|
| ENTITY-TYPE | A1 | Applic, Event, Status, Job |
| ENTITY-NAME | A32 | Name of Entity. |
| ENTITY-IDENTIFIER | A32 | Secondary ident for Entity. |
| ENTITY-FUNCTION | A2 | Function requested |
| RESP | P5 | Response code to PAC. |
| MSG | A78 | Message corresponding to RESP. |
| USER-AREA | A50 | User data for all exits. |

## PACEX001 - PAC Initialization

This user exit is invoked during the initialization of PAC, and allows the user to set up a 50-byte parameter area to be passed throughout the session.

Control must be restored to PAC at the termination of the exit.

### Possible Use

- Pre-initialization logic;
- Additional security;
- Accounting, logging, or tracking statistics.

### Parameter

| Parameter | Format | Description |
|---|---|---|
| USER-AREA | A50 | Modifiable; area for user to use |

## PACEX002 - PAC Termination

This user exit is invoked during the termination of PAC. Any required processing may be done in this routine. The user area, if set up in PACEX001, will be made available for use in PACEX002. Because the user area is passed to PACEX002 using a stack, an INPUT statement is required to access it.

The user will be at ET status on entry into the routine. Control may optionally be returned to PAC at termination of this exit.

### Possible Use

- Additional security;
- Accounting, logging, or tracking statistics;
- "Post-PAC" processing.

### Parameter

| Parameter | Format | Description |
|---|---|---|
| USER-AREA | A50 | Modifiable; area for user to use; passed via STACK Data. |

# PACEX004 - PAC Entity Purge Validation

This user exit  is invoked before a PAC entity is purged, and allows the user to interrogate the entity to be purged before processing by PAC. The purge request may optionally be disallowed.

The user will be at ET status on entry into the routine. Control must be restored to PAC at the termination of the exit.

## Possible Use

This exit can be used to interrogate the entity to be purged before processing by PAC.

## Parameters

| Parameter | Format | Description |
|---|---|---|
| ENTITY-TYPE | A1 | Not modifiable; the output is one of the following entity types:<br>-A Application<br>E Migration event<br>F File translation table<br>J Job<br>L Application status link<br>M Maintenance request<br>S Status<br>T Migration Path |
| ENTITY-NAME | A32 | Not modifiable; the output is the entity name; for example, the name of the status, event, or application. |
| ENTITY-IDENTIFIER | A32 | Not modifiable; the output is a secondary name to identify the entity; for example, the status name of an application status link. |
| ENTITY-FUNCTION | A2 | Not modifiable; the output is the requested function: P Purge entity |
| RESP | P5 | Input; return code. A non-zero return code disallows the purge function for the entity currently being processed. |
| MSG | A78 | Input; user message to be displayed when disallowing the purge function for an entity. |
| USER-AREA | A50 | Modifiable; area for user to use. |

# PACEX005 - Locking Objects at Migration Time

This user exit is invoked before the locking of an object, under the following two conditions:

- For out-migrations (migrations with the to-status either of type (D) or (M)).
- For migrations whose from-status is of type (C) Control.

A non-zero response value will stop the object being locked by PAC and thus excluding the said object from the migration. The user exit is invoked once for every object that is to be locked. If the response value is set to zero (0), PAC will process the object in the normal way.

## Possible Use

When migrating objects out of PAC, the exit allows the user the opportunity to stop certain objects being migrated. If an event had already been created and authorized but the user wanted to stop a certain object from being migrated, without having to modify the event in any way, then this could be achieved with the use of this exit.

## Parameters

| Parameter | Format | Description |
|---|---|---|
| APPL | A32 | Not modifiable; name of the application for the event FM-STAT A32 Not modifiable; origin status name. |
| TO-STAT | A32 | Not modifiable; destination status name. |
| FM-STAT-TYPE | A1 | Not modifiable; origin status type. |
| TO-STAT-TYPE | A1 | Not modifiable; destination status type. |
| EVENT | A32 | Not modifiable; name of the event being processed. |
| OBJ | A32 | Not modifiable; object name. |
| OBJ-TYPE | A4 | Not modifiable; object type |
| OBJ-VERS | N4 | Not modifiable; object version. |
| RESP | P5 | Modifiable; response code. |

## PACEX006 - Compilation Error Verification

This user exit is invoked after the compile processing step of the event is completed when any compilation errors are detected. Normally, PAC would force the event to abnormally terminate; however, this exit may be used to prevent the termination.

If the error number (ERROR-NUM) is set to a non-zero value, it will instruct PAC to terminate processing abnormally as usual. The batch condition code will be 48. The Backout or Recovery functions may be performed normally. If necessary, the next step in the job may invoke a batch job to back out the event. When processing is terminated due to compile errors, this information is written to the event audit report.

If the error number is set to zero (0), PAC will not terminate the event and processing will continue.

Any required processing may be done in the routine. The user will not be at ET status on entry into the routine.

### Possible Use

When migrating objects into PAC from development or maintenance status types, the object list may be an arbitrary list of objects or it may be a "unit of work" comprising a specific set of objects. In the latter case, it may be important for the event to be terminated and subsequently backed out if any of the objects are not successfully compiled.

By issuing a TERMINATE nn statement in the exit, or by leaving the error number as a non-zero value, the user may terminate the compile step of the migration with a specific condition code so that the next step in the JCL/JCS will be executed only if that condition code was received from the compile step. This conditional step may then be a batch command to PAC to back out the event that just failed.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| USER-AREA | A50 | Modifiable; area for user to use |
| EVENT-NAME | A32 | Not modifiable; name of the event being processed |
| APPLICATION | A32 | Not modifiable; name of the application for the event |
| FROM-STATUS | A32 | Not modifiable; origin status name |
| FROM-STATUS-CODE | A1 | Not modifiable; origin status type |
| TO-STATUS | A32 | Not modifiable; destination status name |
| TO-STATUS-CODE | A1 | Not modifiable; destination status type |
| COMPILED-OBJECTS | P8 | Not modifiable; total objects compiled |
| ERROR-OBJECTS | P8 | Not modifiable; total objects not compiled |

| Parameter | Format | Description |
|---|---|---|
| ERROR-MSG | A32 | Modifiable; error message text |
| ERROR-NUM | P5 | Modifiable; error message number |

## PACEX007 - Change Control List Verification

This user exit is invoked during the check-out of objects to and the check-in from a development or maintenance status type. The user has the option to process the information or to reject the request.

**Note:** Since PAC currently does not support message switching facilities, it is recommended that you integrate your own message switching facilities. For example, use Con-nect from User Exit 7 (PACEX007) in the check-out/check-in process of PAC so that all users are correctly informed about check-out/check-in activities.

The parameter list provided with this exit allows you to customize the check-in and check-out processing to your exact requirements.

As each object is being processed for maintenance, the user is given the opportunity to verify the information. A non-zero response will instruct PAC not to accept the specific object for processing.

Control must be restored to PAC after processing in the exit is completed.

Because PACEX007 processing can be complex, it is recommended that you keep the logic in this exit simple by using subprograms to perform specific functions, depending on the FUNCTION or OPTION TYPE parameters.

A detailed explanation of the parameters for PACEX007 is included later in this section.

### Possible Use

The exit may be used to verify the check-out of objects as follows:

- Validate the user and the object being checked out, and optionally disallow the check-out; that is, restrict access to the object.
- Disallow the check-out for a particular object.
- Obtain reports using information about an object being checked out.
- Get a list of other users who also may have previously checked out the same object (concurrent maintenance).
- Validate the user and the object being checked in and optionally disallow the check-in.
- Perform message switching logic to inform users of the checking in or checking out of an object.

- Ensure that all maintenance activities have an associated maintenance request.
- Attach comments or instructions to an object being checked in or out.
- Validate the library where the object is being checked in or out for a user.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| FUNCTION | A1 | Not modifiable; the function may be one of the following: I check-in<br>O check-out |
| OPT-TYPE | A1 | Not modifiable; the option type may be one of the following: blank self (first time call) C closed<br>D duplicate<br>I check-in of object<br>N object not checked out<br>O check-out of object |
| ACC-TYPE | A1 | Not modifiable; the type of access may be one of the following: A access only<br>C check-out request cancelled<br>D development type event called<br>U update |
| APPLICATION | A32 | Not modifiable; name of the application for the object |
| EVENT | A32 | Not modifiable; name of the migration event |
| MAIN-REQ-ID | A20 | Not modifiable; maintenance request ID |
| FROM-STATUS | A32 | Not modifiable; origin status |
| FROM-STATUS-TYPE | A1 | Not modifiable; origin status type |
| TO-STATUS | A32 | Not modifiable; destination status |
| TO-STATUS-TYPE | A1 | Not modifiable; destination status type |
| OBJECT-ID | A32 | Not modifiable; object currently being processed |
| CHECKOUT-VERNO | P5 | Not modifiable; check-out version number |
| CHECKOUT-USER | A8 | Not modifiable; identity of user who checked out the object |
| CHECKOUT-DATE | T | Not modifiable; date and time the object was checked out |
| CHECKOUT-TID | A8 | Not modifiable; identity of terminal from which the object was checked out |
| CHECKIN-VERNO | P5 | Not modifiable; check-in version number for closed logs only |
| CHECKIN-USER | A8 | Not modifiable; identity of user who checked in the object |
| CHECKIN-DATE | T | Not modifiable; date and time the object was checked in |
| CHECKIN-TID | A8 | Not modifiable; identity of terminal from which the object was checked in |
| MNT-LIBRARY | A8 | Not modifiable; destination library |

| Parameter | Format | Description |
|---|---|---|
| MNT-DBID | N5 | Not modifiable; destination DBnr |
| MNT-FNR | N5 | Not modifiable; destination Fnr |
| COMMENT-01 | A60 | Modifiable; comment area |
| COMMENT-02 | A60 | Modifiable; comment area |
| COMMENT-03 | A60 | Modifiable; comment area |
| WORK-AREA | A100 | Modifiable; area for user to use |
| AUD-MSG | A72 | Modifiable; text of message to audit report |
| RESP | N4 | Modifiable; response code |

## Explanation of Parameters

### FUNCTION

When objects are migrated to a (destination) maintenance status type, change control logs containing check-out information are created for each object being migrated. For migrations with an (origin) maintenance status type, the change control log is updated with check-in information for each object checked in from maintenance. No change control logs are created for migrations to or from a development status type. The following options are valid:

I Check-in function being performed on object.

O Check-out function being performed on object.

### OPT-TYPE

Option type specifies the following:

blank Indicates the first time the exit is called for a particular object. At this point, the user may reject the object check-out/check-in based on the name of the object or the user invoking the request. For check-in, the user may wait to decide whether to reject the object until the exit is invoked again with a subsequent call (the exit is invoked with the I option or the N option if the object was not checked out before the check-in request).

C Closed. Indicates that the object being checked in was previously checked out and then checked back in (which closed the check-out log).

D Duplicate object. Indicates that the object being checked out was already checked out to this library with a previous migration event. This could occur when the wrong version was checked out or the version being checked out needs to be refreshed. This situation does not occur with check-ins. It is the user's responsibility to resolve inconsistencies that may occur when multiple versions of the same object are maintained at the same time. Depending on timing and circumstances, programmers may implement the same change in different ways and PAC has no way do determine this.

I In. Indicates that the log information being passed to the exit is for the object currently being checked in. This occurs only on check-ins and gives all of the check-out information about this object. I differs from the (blank) option in that the check-out information is not made available to the user with the (blank) option.

N Never checked out. Indicates that an object now being checked in was never checked out of PAC (for example, a completely new object with no versions in PAC, or a version of a PAC object that was created in development but is being migrated into PAC through maintenance).

O Out. Always indicates that the object is checked out. This may be seen with check-ins or check-outs. Thus, with a check-out, the user is informed about other users who have checked out this same object (but perhaps a different version). For check-ins, the user is informed about the user who still has this same object (but perhaps a different version) checked out. This information is stored in the audit report of the migration event.

### ACC-TYPE

Access type informs the user whether or not the object has been modified after a check-out.

U Update. When the object is checked out, PAC always sets the access type to U (update) and is retained as a U if the object checked out was changed.

A Access. If PAC notes at check-in that the object checked out was not modified, then the access type is set to A (access).

C Cancelled. Occurs when an object was checked out for maintenance, but the request is being cancelled. This is only possible when the user is performing a migration from a maintenance (M) status type to a retire (R) status type.

D Development. This special access type, which can be ignored, occurs when users are migrating objects out of PAC to a development (D) status type. It allows the user to control programming changes between development and maintenance. For example, it may be useful for development to know of changes to objects in a maintenance environment so that the same changes may be made to the objects in the development environment. Where development and maintenance are the same environment, the D option is irrelevant. Alternatively, development may be disallowed if the object is in maintenance.

### APPLICATION

This parameter identifies the application to which the object belongs. In PAC, all activities on objects are always performed at the application level.

### EVENT

This parameter is the migration event being used for the processing of objects. If event naming standards are defined, the event name may help to determine whether or not the checking in or checking out of an object is valid.

### MAIN-REQ-ID

This parameter may be useful to relate a particular migration to a specific problem. All objects being checked in and out will be assigned to that particular maintenance request thus making tracking easier. For example, it may be useful to disallow check-in or check-out of an object unless there is an associated maintenance request.

#### Status Parameters

| | |
|---|---|
| FROM-STATUS | origin status of the migration; that is, where the objects are being migrated from. |
| FROM-STATUS-TYPE | origin status type. |
| TO-STATUS | destination status of the migration; that is, where the objects are being migrated to. |
| TO-STATUS-TYPE | destination status type. |

### OBJECT-ID

This parameter identifies the object currently being processed by PAC.

#### Check-Out Parameters

| | |
|---|---|
| CHECKOUT-VERNO | object version number of the object being checked out. |
| CHECKOUT-USER | user who is checking out the object. |
| CHECKOUT-DATE | date and time the object is being checked out. |
| CHECKOUT-TID | terminal where the object is being checked out. |

#### Check-In Parameters

| | |
|---|---|
| CHECKIN-VERNO | object version number of the object being checked in. |
| CHECKIN-USER | user who is checking in the object. |
| CHECKIN-DATE | date and time the object is being checked in. |
| CHECKIN-TID | terminal where the object is being checked in. |

#### Maintenance Parameters

| | |
|---|---|
| MNT-LIBRARY | maintenance library location to which the object was checked out, or the maintenance library location from which the object is being checked in. For example, if the migration is from production to maintenance, then PAC does not provide the location of production; that is, the origin status. |
| MNT-DBID | database number of the maintenance library. |
| MNT-FNR | file number of the maintenance library. |

#### COMMENT

This parameter identifies the comments passed to the exit. Comments may be modified by the user, for example, to pass messages, reminders, or instructions to someone reviewing change control logs. This information may be set up during the check-out and then used as verification of the object during check-in.

#### WORK-AREA

The WORK-AREA parameter is a 100-byte area that the user may use during the processing of the exit. The work area may be set up when the exit is invoked for the first time (OPT-TYPE blank) and then used each time the exit is invoked for that particular object. PAC does not clear the area until the next object is processed.

#### AUD-MSG

If this parameter is set to a non-blank value, PAC will write the value as a message to the audit report.

### RESP

This parameter tells PAC what to do with the object being processed.

- A zero response indicates that there are no problems and processing can continue.
- A non-zero response instructs PAC to reject the object.

If updates are performed on user-defined files during exit processing, the End Transaction (ET) is performed by PAC. If the user wants to write information to the job listing or to the screen, then this should be done in the exit itself.

## PACEX008 - Application Status Library Validation

This user exit is invoked during the creation of application status links for a particular application. The exit allows the user to validate the specific libraries assigned to each specific application status link.

If the error number (ERROR-NUM) is set to a non-zero value, it will instruct PAC to not accept the specific library being validated. No END/BACKOUT TRANSACTION statements should be issued.

Any required processing may be done in this routine. The user will not be at ET status on entry into the routine.

### Possible Use

Possible uses for this exit are as follows:

- Restrict the assignment of a library for use by a specific application;
- Assign automatic definitions for the value of a library and/or file and/or database for an application when it is linked to a status;
- Override the default settings of the library, database, and file number when an application is linked to a status.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| FUNCTION | A1 | Not modifiable; type of request: A add<br>M modify |
| APPLICATION | A32 | Not modifiable; name of application |
| STATUS | A32 | Not modifiable; name of status |
| STATUS-TYPE | A1 | Not modifiable; type of status |
| LIBRARY | A8 | Modifiable; library for link |
| DBID | N5 | Modifiable; database location of library |
| FNR | N5 | Modifiable; file location of library |
| XREF-FLAG | A1 | Modifiable; XREF usage indicator |
| ERROR-NUM | P5 | Modifiable; response code |
| MSG | A78 | Modifiable; text of message to be given to user |
| USER-AREA | A50 | Modifiable; area for user to use |
| EXTERNAL-USER | A253 | Optional; Input/output |
| EXTERNAL-CODEPAGE | A64 | Optional; Input/output |

## PACEX009 - Duplicate Object Verification

This user exit  is invoked when PAC finds a duplicate object. The duplication is determined by comparing the date-time stamp of the development version with the most recent PAC version of the object.

Normally, PAC ignores duplicate versions of text and copy code object types only. The reason for this is that, while the saved objects may be identical, the cataloged objects may be different. The difference is caused by the external objects (data areas, copy code, views, and rules) used by the compiled object. Since the cataloged and saved objects must always be in sync, a new version must be created. PAC does not check the status of an object's external objects; therefore, it must create a new version of all objects except copy code and text.

The user may override PAC's decision by setting RESP-CODE as follows:

- If RESP-CODE is set to zero, PAC processes the duplicate object;
- If RESP-CODE is set to a non-zero value, PAC ignores the duplicate object.

  The user is at the End Transaction (ET) status on entry to the routine. Control must be returned to PAC when the exit terminates.

The following is the table of corresponding values for object types used by PAC. The value passed by PAC is the short text.

| Code | Short Text | Long Text |
|---|---|---|
| C | Copycd | Copycode |
| T | Text | Text |
| G | Global | Global |
| L | Local | Local |
| A | Param | Parameter |
| P | Progrm | Program |
| S | Subrtn | Subroutine |
| N | Subpgm | Subprogram |
| H | Help | Helproutine |
| M | Map | Map |
| Z | Record | Recording |
| Y | XpertM | Xpert Model |

### Possible Use

Possible uses for this exit are as follows:

- Override PAC to ignore or not ignore the creation of a new version of a particular object;
- Establish different rules of duplication for each application according to object naming conventions, or the type of origin status.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| PAC-STATUS | A1 | Not modifiable; origin status type: D development M maintenance |
| PAC-APPLICATION | A32 | Not modifiable; application for the migration |
| OBJECT | A8 | Not modifiable; object being processed |
| OBJ-TYPE | A15 | Not modifiable; object type |
| OBJECT-DATE | T | Not modifiable; save date of the object |
| RESP-CODE | N4 | Modifiable; response code |

# PACEX010 - Object List Verification

This user exit is invoked during object list processing whenever an entry is processed from the object list of the event. This user exit is valid for all migrations.

As each entry in the object list is passed to the exit, the user has the option to disallow that entry by moving a non-zero response code to the RESP-CODE field. An entry is allowed with a zero response code. Control must be restored to PAC at the termination of the exit.

## Possible Use

PACEX010 is called during the actual reading of the object list before PAC processes each entry on the list; for example:

```
SUBRTN1,S,
HELP*,M,PRODUCTION
```

This allows the user to verify the following:

- Use of range notation;
- Use of references (version or status); these may be disallowed or restricted;
- Use of certain types of objects (perhaps for exclusion);
- Use of itemized objects to ensure that standards are satisfied with naming conventions; that is, that *, <, and > have not been specified for range notations;
- Use of other external subsystems;
- Secure the definition of which objects belong to which applications.

## Parameters

| Parameter | Format | Description |
|---|---|---|
| APPLICATION | A32 | Not modifiable; application name |
| FM-STATUS | A32 | Not modifiable; origin status name |
| TO-STATUS | A32 | Not modifiable; destination status name |
| EVENT | A32 | Not modifiable; migration event name |
| OBJECT | A32 | Not modifiable; object name |
| OBJ-TYPE | A15 | Not modifiable; object type text |
| STAT-VER | A37 | Not modifiable; status/version reference |
| EXIT-AREA | A20 | Not modifiable |
| REDEFINE EXIT-AREA | | |

| Parameter | | Format | Description |
|---|---|---|---|
| | LIBRARY | A8 | Not modifiable; library of the From/To status (see note below) |
| | DBID | N5 | Not modifiable; DBnr of the From/To status |
| | FNR | N5 | Not modifiable; Fnr of the From/To status |
| | RESERVED | A4 | Not used |
| | FROM-STAT | A1 | Not modifiable; From status type |
| | TO-STAT | A1 | Not modifiable; To status type |
| RESP-CODE | | N4 | Modifiable; response code |
| DYN-CODE-OPTION | | A1 | Optional; Output only<br>"S" dynamic source code option is specified |

**Note:** LIBRARY will contain the value of the library for the application status link of either the origin or destination status. If the object is being migrated into PAC, then this will be the library of the development, maintenance, or incorporation status. If the object is being migrated out of PAC, then it will be the library of the destination status.

## PACEX011 - Job Validation

This user exit is called when a job maintenance subfunction is invoked to perform updates or access jobs defined to PAC.

### Possible Use

This exit can be used to implement tracing and customized audit security.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| ENTITY-TYPE | A1 | Not modifiable; the type of entity:<br>"J" job |
| ENTITY-NAME | A32 | Not modifiable; name of the job being processed |
| ENTITY-IDENTIFIER | A32 | Not modifiable; name of new job when ENTITY-FUNCTION is "C" or "R" |
| ENTITY-FUNCTION | A2 | Not modifiable; requested function:<br>"A" add job<br>"C" copy job<br>"D" display job<br>"E" edit JCL<br>"M" modify job<br>"R" rename job<br>"S" select job<br>"U" submit a job |

| Parameter | Format | Description |
|---|---|---|
| RESP | P5 | Modifiable; return code. A non-zero value disallows the requested function for the job currently being processed |
| MSG | A78 | Modifiable; user message to be displayed when disallowing the requested function for the job currently being processed |
| USER-AREA | A50 | Modifiable; area for user to use |

## PACEX012 - Job Submission Verification

This user exit is invoked during JCL submission, and provides the first 10 lines of JCL so that the user may examine them and make any desired changes prior to job submission.

If job processing is to be terminated, then a non-zero response must be returned to PAC.

### Possible Use

This exit can be used to enforce site-specified standards for performing various substitutions (for example, with jobs cards, account information).

### Parameters

| Parameter | Format | Description |
|---|---|---|
| USER-AREA | A50 | Modifiable; area for user to use |
| EVENT-NAME | A32 | Not modifiable; name of the event. This name is available when submitting JCL with the migration event Submit function |
| JCL | A72/1:10 | Modifiable; I/O JCL cards to be examined and optionally modified |
| ERROR-MSG | A78 | Modifiable; user message to be displayed when disallowing the Submit function for the job currently being processed |
| ERROR-NUM | P5 | Modifiable; return code. A non-zero value disallows the Submit function for the job currently being processed |

## PACEX013 - Migration Event Validation

This user exit is invoked whenever any option on the Migration Event Maintenance subfunction is invoked.

### Possible Use

This exit can be used to implement audits and security.

### Parameters

**Note:** The ENTITY-FUNCTION (R), refresh event, cannot be used as an input parameter but if this is set when the "refresh" command is used, then the exit will be invoked.

| Parameter | Format | Description |
|---|---|---|
| ENTITY-TYPE | A1 | Not modifiable; type of entity<br>"E" migration event |
| ENTITY-NAME | A32 | Modifiable; migration event name |
| ENTITY-IDENTIFIER | A32 | Modifiable; new migration event name |
| ENTITY-FUNCTION | A2 | Not modifiable; requested function:<br>"A" add an event<br>"C" copy an event<br>"D" display an event<br>"M" modify an event<br>"R" refresh an event<br>"S" select an event<br>"U" submit an event |
| RESP | P5 | Modifiable; return code. A non-zero value disallows the requested function for the migration event currently being processed |
| MSG | A78 | Modifiable; user message to be displayed when disallowing the requested function for the migration event currently being processed |
| USER-AREA | A50 | Modifiable; area for user to use |

## PACEX014 - Status Validation

This user exit is invoked when any option on the Status Maintenance subfunction is invoked.

### Possible Use

This exit can be used to implement audits and security.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| ENTITY-TYPE | A1 | Not modifiable; type of entity<br>"S" status |
| ENTITY-NAME | A32 | Not modifiable; status name |
| ENTITY-IDENTIFIER | A32 | Not used |
| ENTITY-FUNCTION | A2 | Not modifiable; requested function:<br>"A" add a status definition<br>"D" display a status definition<br>"L" link status to applications<br>"M" modify status definition<br>"S" select status |
| RESP | P5 | Modifiable; return code. A non-zero value disallows the requested function for the status currently being processed |
| MSG | A78 | Modifiable; user message to be displayed when disallowing the requested function for the status currently being processed |
| USER-AREA | A50 | Modifiable; area for user to use |

## PACEX015 - Application Validation

This user exit is invoked when an Application Maintenance subfunction is invoked to perform updates or access applications defined to PAC.

### Possible Use

This exit can be used to implement tracing and customized audits and security.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| ENTITY-TYPE | A1 | Not modifiable; the type of entity:<br>"A" application |
| ENTITY-NAME | A32 | Not modifiable; the name of the application being processed |
| ENTITY-IDENTIFIER | A32 | Not modifiable; new application name when ENTITY-FUNCTION is C |

| Parameter | Format | Description |
|---|---|---|
| ENTITY-FUNCTION | A2 | Not modifiable; the requested function:<br>"A" add application<br>"C" copy application<br>"D" display application<br>"L" link application to a status<br>"M" modify application<br>"S" select application |
| RESP | P5 | Modifiable; return code. A non-zero value disallows the requested function for the application currently being processed |
| MSG | A78 | Modifiable; user message to be displayed when disallowing the requested function for the application currently being processed |
| USER-AREA | A50 | Modifiable; area for user to use |

## PACEX016 - File Translation Table Validation

This user exit is invoked whenever any option on the FTT Maintenance subfunction is invoked.

### Possible Use

This allows the user to implement audits and security.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| ENTITY-TYPE | A1 | Not modifiable; the type of entity:<br>"F" file translation table |
| ENTITY-NAME | A32 | Not modifiable; FTT name |
| ENTITY-IDENTIFIER | A32 | Not modifiable; new FTT name when ENTITY-FUNCTION is C |
| ENTITY-FUNCTION | A2 | Not modifiable; the requested function:<br>"A" add FTT<br>"C" copy FTT<br>"D" display FTT<br>"M" modify FTT<br>"S" select FTT |
| RESP | P5 | Modifiable; return code. A non-zero value disallows the requested function for the FTT currently being processed |
| MSG | A78 | Modifiable; user message to be displayed when disallowing the requested function for the FTT currently being processed |
| USER-AREA | A50 | Modifiable; area for user to use |

## PACEX017 - Application Status Link Validation

This user exit is invoked whenever any option on the Migration Paths subfunction is invoked.

### Possible Use

This exit can be used to implement audits and security.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| APPLICATION | A32 | Not modifiable; the application name |
| FM-STATUS | A32 | Not modifiable; origin status name |
| TO-STATUS | A32 | Not modifiable; destination status name |
| FUNCTION | A2 | Not modifiable; the requested function:<br>"A" add migration path<br>"D" display migration path<br>"M" modify migration path<br>"S" select migration path |
| RESP | P5 | Modifiable; return code. A non-zero value disallows the requested function for the migration path currently being processed |
| MSG | A78 | Modifiable; user message to be displayed when disallowing the requested function for the migration path currently being processed |
| USER-AREA | A50 | Modifiable; area for user to use |

## PACEX018 - Object Version Reporting

This user exit  is invoked whenever any option on the Versioned Objects Reporting subfunction is invoked.

### Possible Use

This exit allows the user to implement audits and security. For example, access to certain objects belonging to certain applications may be restricted to specific users.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| ENTITY-TYPE | A1 | Not modifiable; the type of entity:<br>"V" object version |
| ENTITY-NAME | A32 | Not modifiable; application name |
| ENTITY-IDENTIFIER | A32 | Not modifiable; object name |
| ENTITY-FUNCTION | A2 | Not modifiable; the requested function:<br>"D" display versioned object<br>"H" hardcopy versioned object<br>"S" select versioned object<br>"U" list versioned objects with no application status<br>"V" display versioned object history data |
| RESP | P5 | Modifiable; return code. A non-zero value disallows the requested function for the object version currently being processed |
| MSG | A78 | Modifiable; user message to be displayed when disallowing the requested function for the object version currently being processed |
| USER-AREA | A50 | Modifiable; area for user to use |

## PACEX019 - Maintenance Request Validation

This user exit is invoked whenever ENTER is pressed when using a Maintenance Request subfunction.

### Possible Use

This exit can be used as follows:

- To move objects in and out of PAC;
- To automate modifications to the maintenance request;
- To retrieve information from an external problem-tracking system;
- To perform audits or any automated processing, especially with an external problem-tracking system.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| REQUEST-ID | A32 | Not modifiable; the maintenance request ID |
| REQUEST-ID-2 | A32 | Not modifiable; the new maintenance request ID when the FUNCTION is C (copy) |
| FUNCTION | A1 | Not modifiable; the requested function:<br>"A" add request ID<br>"C" copy request ID<br>"D" display request ID<br>"L" list by request ID<br>"M" modify request ID<br>"S" select request ID |
| PROBLEM-SUBJECT | A60 | Modifiable; subject of the problem |
| PROBLEM-TYPE | A8 | Modifiable; type of the problem |
| PRIORITY | N3 | Modifiable; priority |
| STATUS | A12 | Modifiable; status defined to PAC using Table Maintenance in administrator functions |
| ACTION | A12 | Modifiable; action defined to PAC using Table Maintenance in administrator functions |
| CONTACT-NAME | A32 | Modifiable; name of the contact person who reported or is responsible for the maintenance request |
| CONTACT-TEL | A20 | Modifiable; telephone number of the contact person |
| ERROR-NUM | P5 | Input; return code. A non-zero code disallows the requested function for the maintenance request currently being processed. |
| MSG | A78 | Input; user message to be displayed when disallowing the requested function for the maintenance request currently being processed |
| USER-AREA | A50 | Modifiable; area for user to use as set up by the PAC initialization exit |

## PACEX020 - Maintenance Request Processing Validation

This user exit is invoked when a Maintenance Request subfunction is invoked.

### Possible Use

This exit can be used to implement tracing, audits, and/or customized security.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| ENTITY-TYPE | A1 | Not modifiable; the type of entity:<br>"M" maintenance request |
| ENTITY-NAME | A32 | Not modifiable; the name (ID) of the maintenance request being processed. The first 20 characters are valid. |
| ENTITY-IDENTIFIER | A32 | Not modifiable; the new maintenance request name (ID) when ENTITY-FUNCTION is C. The first 20 characters are valid. |
| ENTITY-FUNCTION | A2 | Not modifiable; the requested function:<br>"A" add request ID<br>"C" copy request ID<br>"D" display request ID<br>"L" list by request ID<br>"M" modify request ID<br>"S" select request ID |
| RESP | P5 | Input; return code. A non-zero code disallows the requested function for the maintenance request currently being processed. |
| MSG | A78 | Input; user message to be displayed when disallowing the requested function for the maintenance request currently being processed |
| USER-AREA | A50 | Modifiable; area for user to use |

## PACEX021 - Maintenance Request Name Verification

This user exit is invoked whenever any modification of the Maintenance Request name is performed during migration event maintenance.

### Possible Use

This exit allows the user to implement audits and security. It can be used to ensure the specification of a maintenance request whenever a migration event is created.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| MAINTENANCE-REQUEST | A32 | Modifiable; name (ID) of the maintenance request |
| EVENT | A32 | Input; name of the migration event |
| APPLICATION | A32 | Input; name of the event's application |
| ORIGIN-STATUS | A32 | Input; name or the event's origin (From) status |
| DESTINATION-STATUS | A32 | Input; name of the event's destination (To) status |
| RESP | P5 | Input; return code. A non-zero code disallows the requested function for the maintenance request currently being processed |
| MSG | A78 | Input; user message to be displayed when disallowing the requested function for the maintenance request currently being processed |
| USER-AREA | A50 | Modifiable; area for user to use |

## PACEX022 - Object Version Used-By-List Reporting

This user exit is invoked whenever subordinates (maps, data areas, or copy code) in use by another object(s) are migrated from development or maintenance status types.

For objects compiled with the steplib option, this user exit identifies the application to which the object belongs so that when the subordinate changes, you can obtain a complete list across all applications of all objects that use that subordinate.

### Possible Use

With Natural, objects using subordinates need not be recompiled until the objects affected by the subordinate change; this user exit can be used to warn the user that this change has taken place.

It can be used as well to write the list of affected objects to a work file for use in the generation of a subsequent migration event, thus automating the change to all objects using a particular subordinate. The work file dataset created by this exit can be used as input to the Generate function that creates another migration event, or it can be used to create the object list for an existing event.

Alternatively, the exit may be used to write the list of affected objects to a print file so that reports can be created that satisfy user requirements.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| EVENT | A32 | Input; name of the migration event |
| APPLICATION | A32 | Input; name of the object's application |
| ORIGIN-STATUS | A32 | Input; name of the event's origin status |
| DESTINATION-STATUS | A32 | Input; name of the event's destination status |
| OBJECT | A32 | Input; name of the object |
| TYPE | A4 | Input; object type |
| VERSION | N4 | Input; object version |
| USED-BY-OBJECT | A32 | Input; name of the used-by object; that is, the object that uses the OBJECT as a subordinate object |
| USED-BY-TYPE | A4 | Input; object type of the used-by object |
| USED-BY-VERSION | N4 | Input; object version of the used-by object |
| USED-BY-APPLICATION | A32 | Input; application of the used-by object |
| USER-AREA | A50 | Modifiable; area for user to use |

## PACEX023 - Object Verification for Autoexpand

This user exit is called before a particular object is expanded. PAC passes the object name, type, version, and the Expand option to the exit.

### Possible Use

The user may change the expand option for the particular object by modifying the Expand option value; or suppress the expansion of the object by returning a non-zero value in the return code field.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| EVENT | A32 | Input; name of the migration event |
| APPLICATION | A32 | Input; name of the application |
| ORIGIN-STATUS | A32 | Input; name of the origin status |
| DESTINATION-STATUS | A32 | Input; name of the destination status |
| OBJECT | A32 | Input; name of the object |
| TYPE | A4 | Input; object type as specified in the object list |
| VERSION | N4 | Input; object version |

| Parameter | Format | Description |
|---|---|---|
| EXPAND-OPTION | A1 | Modifiable;<br>"C" subordinate objects (compile time)<br>"E" subordinate and referenced objects<br>"N" no expansion<br>"R" referenced objects (run time)<br>"U" used-by objects |
| RETURN-CODE | P5 | Modifiable;<br>"zero" expand object<br>"non-zero" do not expand |
| USER-AREA | A50 | Modifiable; area for user to use |

## PACEX024 - Object Participation in Migration Reporting

This user exit lets you obtain a list of objects which come up for an in-migration or out-migration as a result of the interpretation of the migration list. It also is able to list objects whose in-migration or out- migration is still attempted after the restrictions imposed on the migration by various switches, applymods, and user exits have been taken into account.

### Possible Use

The exit may be used during both Natural and Predict migrations.

The exit does not let you prohibit the migration of selected objects.

The exit is not used at any retirement or archiving migration.

The fact that an object has been cleared for the migration does not necessarily imply that the object makes it into the destination of the migration. For example, the compiling, which is part of an in-migration from a development or maintenance status takes place after the clearing. Likewise, all PAA activities formally follow the clearing and cannot be monitored through this exit alone.

In some cases, during the course of an out-migration, a PAC object may acquire an object-status link even though it is not migrated: for example, a data area listed for out-migration with Applymod 8 off. Such objects are not reported as cleared for out-migration.

### Parameters

All fields are used only to pass values to PACEX024. Any modifications of these values in PACEX024 or in modules it may call are ignored.

| Parameter | Format | Description |
|---|---|---|
| FUNCTION | A2 | Not modifiable. The function may be one of the following:<br>IB = an object is being reported as selected for an in-migration as a result of the interpretation of a migration list entry.<br>IA = an object is being reported as cleared for an in-migration.<br>EB = an object is being reported as selected for a out-migration as a result of the interpretation of a migration list entry or because of its participation in the in-migration part of the transmigration of which the current emigration is the latter part.<br>EA = an object is being reported as cleared for a out-migration. |
| EVENT | A32 | Not modifiable. Name of the migration event. |
| APPLICATION | A32 | Not modifiable. Name of the Application. |
| ORIGIN-STATUS | A32 | Not modifiable. Name of the from-status. |
| ORIGIN-TYPE | A1 | Not modifiable. Is the one character code of the type of the origin status of the migration. The possible values of ORIGIN-TYPE are:<br>'C' for 'CONTROL<br>''D' for 'development',<br>'I' for 'incorporation',<br>'M' for 'maintenance',<br>'P' for 'production',<br>'T' for 'test'. |
| DESTINATION-STATUS | A32 | Not modifiable. Name of the to-status. |
| DESTINATION-TYPE | A1 | Not modifiable. Is the one character code of the type of the destination status of the migration. The possible values of DESTINATION-TYPE are the same as those of ORIGIN-TYPE except 'I'. |
| OBJECT | A32 | Not modifiable; Name of the object being reported. |
| TYPE | A4 | Not modifiable. Is the code of the type of the object being migrated |
| VERSION | N4 | Not modifiable. Is the version number of the object being migrated. If FUNCTION = 'IB', it is 0; if FUNCTION = 'IA', it carries the version number of the PAC object that will be created if the in-migration is completed. |

## PACEX025 - Migration Event Names

This user exit allows the user to implement the routines for generating and verifying event names. It is invoked when function code A (Add) or C (Copy) is selected from the Migration Event Maintenance menu before the event name (entered by the user) is verified by PAC.

Since this exit is called before validation of the migration event name, the user may optionally leave the Event or the New Event field blank.

### Possible Use

This user exit allows the user to implement the routines for automatically generating and verifying event names. It may be used to perform audit or security functions on existing events or it may be used to restrict the creation of an event.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| ENTITY-TYPE | A1 | Not modifiable; the type of entity<br>"E" migration event |
| ENTITY-NAME | A32 | Modifiable; name of the migration event. |
| ENTITY-IDENTIFIER | A32 | Modifiable; new name of the migration event when the ENTITY-FUNCTION is C |
| ENTITY-FUNCTION | A2 | Not modifiable; requested function:<br>"A" add an event<br>"C" copy an event |
| RESP | P5 | Modifiable; return code. A non-zero value disallows the requested function for the migration event currently being processed |
| MSG | A78 | Modifiable; user message to be displayed when disallowing the requested function for the migration event currently being processed |
| USER-AREA | A50 | Modifiable; area for user to use |

## PACEX026 - Object List Selection for Archiving

This user exit interface provides criteria for selecting objects to be archived. The selected objects are added to the object list of an existing migration event. Only the selection criteria for archiving objects for a particular application are identified. User routine UMENARCH verifies the actual objects selected.

The exit is invoked using the ARCEVENT or Generate Archive List functions.

**Note:** If an event is used, it must be an existing event; it must have a migration path Control to Archive; it must be in a pending or validated state; and its object list will be deleted.

### Possible Use

This exist may be used to set up or override the archive selection criteria parameter values.

### Parameters

| Parameter | | Format | Description |
|---|---|---|---|
| FUNC | | A1 | Not modifiable; indicates when to select objects to be archived:<br>"A" after the exit is invoked<br>"B" before the exit is invoked |
| REDEFINE-OPTS | | A8 | Options for archive processing |
| | 1. DEF-TYPE | A1 | Modifiable; when FUNC is B (before), PAC can use either:<br>"A" the archive selection criteria defaults from the application<br>"U" the parameter defaults passed by the user exit |
| | 2. SHOW-MAP | A1 | Modifiable; show map for input (Y) or get parameters from exit (N):<br>"Y" show an input map to use when overriding parameter values passed by the exit<br>"N" do not show an input map |
| | 3. ARCEVENT | A1 | Modifiable; A = invoked by Archive; R = invoked by Retire Event |
| TO-STATUS | | A1 | Modifiable; A = Archive; R = Retire |
| APPLICATION | | A32 | Name of application to be processed |
| EVENT | | A32 | Name of migration event to be used if the object list must be created |
| MIN-VERSION | | P2 | Number of valid versions to be exempted from archiving; number begins with most recent; default is 3; minimum of three objects must remain. |
| RETENT-DAYS | | P3 | Age of object, in days,before it can be selected for Archiving/Retiring; Zero (0) indicates no data checking |
| STATUS-CHECK | | A1 | Modifiable; used in conjunction with STATUS-LIST when an object in a status is selected:<br>"Y" the default; ignore the selection of an object for archiving in a STATUS-LIST status<br>"N" archive the selected object |
| STATUS-LIST | | A6 | Modifiable; if STATUS-CHECK is Y, this parameter contains the list of status types in which an object selected for archiving is ignored. The default list is as follows:<br>"P" production (any object in a production status is ignored)<br>"M" maintenance<br>"T" test<br>"A" Archive; that is, objects not yet purged |
| RESP-CODE | | N4 | Not used |

## PACEX027 - User Substitution Parameter Verification

This user exit is called whenever a batch job with user substitution parameters is submitted from PAC.

### Possible Use

PAC jobs may contain substitution parameters. This exit may be called before and after the substitution parameters are specified; thus the exit may be used to specify the default substitution parameters. These parameters will then be displayed to the user according to normal job substitution functionality; the user may then change them. Once the user has made the substitutions, the parameters may once again be passed to the exit for validation.

The user may need to perform additional processing (such as additional database calls or input parameters) within this exit to derive the values for the substitution parameters.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| FUNC | A1 | Not modifiable; select objects to be archived before (B) or after (A) the exit is invoked |
| SUBS-CHAR | A1 | Not modifiable; JCL substitution character |
| SUBS-PARM | A32/1:10 | Not modifiable; names of substitution variables |
| SUBS-VALU | A32/1:10 | Modifiable; values of substitution variables |
| SUB1 | P3 | Input; parameter in error |
| USER-AREA | A50 | Modifiable; area for user to use |
| RESP | P5 | Input; error code |
| MSG | A78 | Input; error message |

## PACEX028 - Audit Report Message Verification

This user exit is invoked during migration event processing, usually when a message is being written to the audit report. This exit returns a non-zero response so that the user may optionally suppress the writing of an audit message to the audit report.

Any required processing may be performed in this routine; however, because the user may not be at ET status, control must always be returned to PAC to complete any internal processing.

**Note:** The user should not write to the Natural work source area. That is, do not use the DEFINE PRINTER OUTPUT 'SOURCE' option.

### Possible Use

In most cases, the messages written by the migration process invoke PACEX028 before they are written either to the audit report or to the printer. This allows the user to perform the following:

- Instruct PAC to suppress the writing of the message to the audit report;
- Print for the migration message, including headings;
- Write additional messages to the report listing (not the actual audit report);
- Change a message before it is written to the audit report;
- Check for errors (E), cautions (C), and warnings (W); and optionally write these to an exception report, for example, CMPRT01;
- Change the attributes of the message (for example, color, format).

### Parameters

| Parameter | Format | Description |
|---|---|---|
| EVENT-NAME | A32 | Not modifiable; migration event name |
| APPLICATION | A32 | Not modifiable; application name |
| FROM-STATUS | A32 | Not modifiable; origin status name |
| FROM-STATUS-CODE | A1 | Not modifiable; origin status type |
| FROM-LIBRARY | A8 | Not modifiable; origin status library name |
| FROM-DBID | N5 | Not modifiable; origin status library database number |
| FROM-FNR | N5 | Not modifiable; origin status library file number |
| TO-STATUS | A32 | Not modifiable; destination status name |
| TO-STATUS-CODE | A1 | Not modifiable; destination status type |
| TO-LIBRARY | A8 | Not modifiable; destination status library name |
| TO-DBID | N5 | Not modifiable; destination status library database number |
| TO-FNR | N5 | Not modifiable; destination status library file number |
| PRINT-IT | A1 | Modifiable; N Suppress = Print |
| ERROR-MSG | A78 | Not modifiable; error message text |
| ERROR-NUM | P5 | Modifiable; error message number |

### How to Use PACEX028

When the exit is invoked, do the following to suppress the message:

1. Perform any special validation (for example, batch versus online processing, application, migration path).
2. Set the response code ERROR-NUM to a non-zero value.
3. Override the message, replace the contents of the ERROR-MSG field.

The format of all PAC error messages is as follows:

```
PACnnnn: (t) message-text
```

where:

| | |
|---|---|
| nnnn | is the message number |
| (t) | is the message type: |
| | E (errors) |
| | C (caution) |
| | W (warning) |
| | A (audit) |
| | I (information) |
| message-text | is the actual message |

**Note:** Users should not write to the Natural work source area; that is, do not use the DEFINE PRINTER OUTPUT 'SOURCE' option.

The user may or may not be at ET status and no ET or BT should be issued from the exit; PAC does this.

### Creating an Exception Report

An exception report for all errors can be created as follows:

```
FORMAT  (2)  LS=80 PS=0
IF ERROR-MSG = MASK (.........'(E)'    /* Is this an error message?
OR ERROR-MSG = MASK (.........'(C)'  /* Is this a caution message?
       WRITE (2) ERROR-MSG               /* Yes, write exception report to CMPRT02
END-IF
```

### Color-Coded Reports

With color terminals, messages can be color-coded for reports of online migrations. For example:

| Message Type | Color |
|---|---|
| Audit messages | Turquoise |
| Warnings | Highlighted turquoise |
| Cautions | Highlighted yellow |
| Errors | Highlighted red |

Alternatively, dynamic variables can be used for color-coding the message. For example:

```
.
.
.
REDEFINE ERROR-MSG
  MSG-PREFIX  (A8)
   MSG-DYN1  (A1)
  MSG-TYPE  (A3)
   MSG-DYN2  (A1)
.
.
.
MOVE '|' TO MSG-DYN1
MOVE '¬' TO MSG-DYN2
DECIDE ON FIRST VALUE OF MSG-TYPE
  VALUE '(A)' PRINT ERR-MSG  (CD=TU DY=|TU¬)
  VALUE '(W)' PRINT ERR-MSG  (CD=TU DY=|TUI¬)
  VALUE '(C)' PRINT ERR-MSG  (CD=TU DY=|YEII¬)
  VALUE '(E)' PRINT ERR-MSG  (CD=TU DY=|REI¬)
  ANY              MOVE 4 TO ERR-NUM       /*suppress message in PAC
  NONE            PRINT ERR-MSG  (CD=BL DY=|BLI¬)
```

**Note:** The PAC audit report stores only the first 72 bytes of any message.

## PACEX029 - Test Production Emigration

PACEX029 (if switched on) is called at the application of an FTT to a module being emigrated to a test or production location: once for each reference to a file in the module to which the FTT is being applied. FTTs are applied to modules migrated to production deployments by PAA, not by PAC. Therefore the switch which determines whether PACEX029 is to be called at an emigration to production is in the relevant FPAA, not in the ACF: it is the switch of PAA user exit 6.

The name of the subprogram to be called is still PACEX029, and it is the steplib chain of SYSPAA that determines which libraries and in what order are to be searched for the subprogram. PACEX029

is not called for modules which have no references to ADABAS files. PACEX029 is called regardless of whether the reference to a file is within the scope of any entry of the FTT being applied.

PACEX029 is called only once for each reference even if the reference is within the scope of several entries of the FTT and all of them are applied; in the latter case the values of NEW-DBNR and NEW-FNR are the net resulting ones. Only the values of MSG and RESP can be usefully modified by PACEX029.

If RESP is set at 0, then the processing will continue as it would with the user exit off. If RESP is set at 1, then 'PAC7382: (I) ' or 'PAA0087: (I) ' will be displayed or printed followed by [MSG], and the processing will continue. If RESP is set at anything else, then a message like that produced when RESP EQ 1 will be displayed or printed, and the processing will terminate, in batch with RC=055.

If an emigration to test is terminated with PACEX029, then the user should reckon with some migrating objects having been copied to the destination location(s) with the FTT applied and some not having been copied there - previous versions, if any, surviving.

For a workfile migration these two classes of objects will account for all migrating objects, the object that has caused the termination not having been copied. For a non-workfile migration the object that has caused the termination will be an only exception: it will have been copied untranslated.

A PACEX029-terminated emigration to test, whether UNDOne or RELEASEd, will leave the test location(s) with an inconsistent set of modules. Therefore users are advised not to immediately unlock PACEX029-terminated events to test.

### Parameters

| Name | Format | I/O |
|---|---|---|
| FTT | A32 | in |
| OBJECT | A32 | in |
| OBJ-LIBRARY | A8 | in |
| OBJ-DBNR | N5 | in |
| OBJ-FNR | N5 | in |
| OLD-DBNR | N5 | in |
| OLD-FNR | N5 | in |
| NEW-DBNR | N5 | in |
| NEW-FNR | N5 | in |
| MSG | A72 | out |
| RESP | N4 | out |

## How to Use PACEX029

### To obtain a state functionally equivalent to the pre-emigration state:

1. Create an event from the test status to RETIRE.
2. Use GENLIST L, (N,N) to create its migration list.
3. Create an event from CONTROL to the test status.
4. Use GENLIST L, (H,Y) to create its migration list
5. RELEASE the locked emigration event.
6. Authorise and submit the just created retirement event.

**Tip:** If the original, PACEX029-terminated event was a COPY (not MOVE) from a maintenance status, and applymod 8 or 19 was off, then it may be necessary to drop some entries from the migartion list of the retirement event; this should be done at the validation of the retirement event in response to messages complaining of the absence of objects in the origin status.

### To keep in the test location(s):

To keep in the test location(s) the objects already copied there and successfully translated and to resume the emigration from the object that has caused the termination (presumably, after having corrected the FTT or assigned a different one):

1. Create an event from the test status to RETIRE.
2. Use GENLIST L, (N,N) to create its migration list.
3. Delete the part of the list from, and including, the entry referring to the object that has caused the termination to the end of the list.
4. Create an event from CONTROL to the test status.
5. Use GENLIST L, (O,Y) to create its migration list.
6. Delete the part of the list from its beginning to, but excluding, the entry referring to the object that has caused the termination.
7. RELEASE the locked emigration event.
8. Authorise and submit the just created retirement event.

**Tip:** If the original, PACEX029-terminated event was a COPY (not MOVE) from a maintenance status, and applymod 8 or 19 was off, then it may be necessary to drop some entries from the migartion list of the retirement event; this should be done at the validation of the retirement event in response to messages complaining of the absence of objects in the origin status.

**Tip:** If an emigration to production is terminated with PACEX029, then the user should reckon with all migrating objects having been copied (invisible) to the destination location(s) with the FTT(s) applied to some and not applied to the others. The object which has caused the termination will not have been translated. The PAA job will be Pending and its objects Scheduled; the previous versions of the objects, if any, will remain in place. The job will be fit only for purging. A PACEX029-terminated emigration to production should be UNDOne rather than RELEASEd; this, together with the purging of the PAA job, will return the system to almost the pre-emigration state (apart from some historical object - status links). Users are advised against trying to retire from production objects whose emigration has been terminated with PACEX029, especially if the PAA job has already been purged. There is no method of completing a PACEX029-terminated emigration to production without re-emigrating the objects successfully copied to and translated in the destination location(s).

**Note:** Of the PACEX029-generated messages only 7382s will make it into the audit report and only if the emigration is not a workfile one.

## PACEX030 - Event Authorization Verification

This user exit allows the user to implement additional controls for event authorization before it is actually authorized. The exit is invoked before and after authorization begins. Any required processing may be performed in this routine. The user will be at End Transaction (ET) status on entry into the routine.

Switch for Authorization: Alignment events (from CONTROL to CONTROL) and events of migrations from neighbour statuses to CONTROL provide a new tripositional switch - Catalogue - to be set at authorisation time. The field COPY-MOVE-INCLUDE is designated for double purposes. PACEX038 is PACEX030 enhanced.

### Possible Use

- Automatic override of parameters for certain authorizations;
- Additional audit of the security checking for authorizations;
- Set up of parameters not previously defined in the migration path.

## Parameters

| Parameter | Format | Description |
|---|---|---|
| FUNCTION | A1 | Input; indicates when the exit is to be invoked:<br>"A" after authorization begins<br>"B" before authorization begins |
| EVENT | A32 | Input; event name |
| APPLICATION | A32 | Input; event's application name |
| FROM-STATUS | A32 | Input; event's origin status name |
| TO-STATUS | A32 | Input; event's destination status name |
| JOB | A32 | Modifiable; job name for batch processing |
| EXPAND-OPT | A1 | Modifiable; Expand option:<br>"C" subordinate objects (compile time)<br>"E" subordinate and referenced objects<br>"N" no expansion<br>"R" referenced objects (run time)<br>"U" used-by objects |
| EXPAND-STATUS | A1 | Modifiable; name of the status to be used as the reference of expanded objects |
| BATCH-ONLINE | A1 | Modifiable;<br>"B" run the migration event in batch<br>"O" run the migration event online |
| COPY-MOVE | A1 | Modifiable;<br>"C" copy objects<br>"I" include objects<br>"M" move objects |
| WORKFILE-USAGE | A1 | Modifiable;<br>"Y" use work file<br>"N" do not use work file |
| APPLYMODS | A50 | Modifiable; applymod default settings (A/D/Y/N) are still enforced |
| SCHED-DATE | T | Modifiable; date/time the migration event is scheduled to run |
| TRANSLATION-TABLE | A32 | Modifiable; name of the file translation table to be used for the migration event |
| MSG-NO | P5 | Modifiable; error message number |
| MSG | A78 | Modifiable; error message text |
| USER-AREA | A50 | Modifiable; area for user to use |

## PACEX031 - Passing CATALL Parameters to PAC

This exit is invoked prior to CATALL during the migration of objects from development or maintenance. It allows you to pass CATALL parameters to PAC for use during the compile.

If "NOREN" is specified, the following CATALL will not renumber the lines of the sources which it compiles; it will renumber them if "NOREN" is not specified. If "NOSCROLL" is specified, the following CATALL will not scroll the list of compiled objects which it produces; it will scroll the list if "NOSCROLL" is not specified.

| Parameter | Format | Description |
|---|---|---|
| APPLICATION | A32 | Input; name of the application |
| FROM-STATUS | A32 | Input; event's origin status name. |
| TO-STATUS | A32 | Input; event's destination status name. |
| PARM | A50 | Modifiable; "U" or "NOREN" or "NOSCROLL" or "NOREN,NOSROU" |

## PACEX032 - Rejecting Objects for Object List

This exist is invoked for each object selected by PAC during the selection of objects for the Object List from the Object List Editor. It is used to reject objects based on a moving non-zero value for the MSG-NO parameter (Natural Optimizer Compiler).

| Parameter | Format | Description |
|---|---|---|
| EVENT | A32 | Input; the name of the event. |
| APPL | A32 | Input; the application of the event. |
| FM-STAT | A32 | Input; the origin of the event. |
| TO-STAT | A32 | Input; the destination of the event. |
| OBJ | A8 | Input; the name of the object. |
| OBJ-TYPE | A4 | Input; the type of the object. |
| OBJ-VERS | N4 | Input; the object's version number if the object resides in CONTROL compartment, 0 otherwise. |
| DATASET | A54 | Input; the name of the PDS from which the foreign object would be immigrated or deleted. Blank if the object is not foreign or the origin of the event is not of type incorporation, development, test, or maintenance or the event does not involve migration and is not of the type "retirement". |
| RESP | P5 | Modifiable; if not equal to 0, the object is rejected. |
| #URL | A253 | Filled only if FM-STAT type is External.<br>Optional; URL of the node/file |

| Parameter | Format | Description |
|---|---|---|
| #NODE | L | Filled only if FM-STAT type is External.<br>Optional; true if the URL is a node |

## PACEX033 - Activity Authorisation Exit

This exit is a subprogramm that is used if PAC runs under security. To work under security, one has to set the Protection field within the System Defaults to 'Y'es, keeping the FSEC DBnr at 0 and the Fnr fields on the same defaults, by switching on the user exit 33.

PACEX033 will then be called whenever PAC has to ascertain that the user has the appropriate permissions for the activity he is trying to start. It will also be repeatedly called at the building time of lists whose entries refer to entities each of which may be differently accessible to the user.

**Note:** A sample PACEX033 and a parameter data area, PACEXA33, are provided in SYSPACUS.

| Parameter | Format | Description |
|---|---|---|
| USERID | A8 | User ID for which permissions are sought. |
| ACF-DBNR | N5 | Numbers of the ACF containing the entities for which permissions are sought. |
| ACF-FNR | N5 | Numbers of the ACF containing the entities for which permissions are sought. |
| KIND | A1 | Possible values are:<br><br>A: if application permissions are sought<br>J: if JCL text permissions are sought<br>T: if FTT permissions are sought |
| NAME | A32 | Contains the name of the application, JCL text or FTT. |
| QUEST | L | Contains TRUE in those cases when the elements which correspond to the permissions are sought. Here is the table of the various permissions which can be granted or denied for each (user,application), (user,JCL text), (user,FTT) pair. |

| Quest | Application | JCL Text | FTT |
|---|---|---|---|
| 0 | reading | reading | reading |
| 1 | adding | adding | adding |
| 2 | modifying | modifying | modifying |
| 3 | deleting | deleting | deleting |
| 4 | event adding | submitting | |
| 5 | event modifying | | |
| 6 | event submitting | | |
| 7 | event deleting | | |

| | | |
|---|---|---|
| QUEST | (0:7) | |

| Parameter | Format | Description |
|---|---|---|
| QUESTA | A1 (0:7) | |
| QUEST | | |
| QUESTB | B1 (0:7) | |
| PERM | L (0:7) | |
| PERM | | |
| PERMA | A1 (0:7) | |
| PERM | | |
| PERMB | B1 (0:7) | |
| MSG | A72 | |
| RESP | N4 | |

## PACEX034 - Direct Command User Exit

This exist is passed by the command line in chunks of alphanumeric 32 blocks.This exit is called before the validation of the command line. Any mocifications to the commands will be validated.

### Possible Use

PACEX0034 can be used to pass site-specific syntax to PAC, for example ADD EVENT my be entered as CREATE LIST.

| Parameter | Format | Description |
|---|---|---|
| COMMAND | A32/36 | Each command as entered. |
| MSG | A78 | Message. |
| RESP | P5 | Response code. |

## PACEX035 - SCANOBJ Utility

The scan utility exit, User Exit PACEX035, is invoked with the SCANOBJ utility for every object to be scanned in PAC.

Because SCANOBJ makes use of windows for online processing, caution should be used when issuing WRITE, PRINT, DISPLAY, or INPUT statements from within the user exit.

### Possible Use

PACEX035 allows the user to perform security checking for each request and then for each object at the time the SCANOBJ utility is invoked.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| FUNC | A1 | Not modifiable;<br>"I" the initial call for allowing a specific scan request<br>"S" the scan call for a specific object being called |
| APPLIC | A32 | Not modifiable; application name |
| OBJECT | A8 | Not modifiable; Natural object name |
| STATUS | A32 | Not modifiable; status name (set up by user). If empty, control will be used. |
| VERSION | N4 | Not modifiable; object version number |
| OBJ-TYPE | A15 | Not modifiable; object type |
| RESP | N4 | Input. Response code. If set not equal to 0, the object will be skipped for FUNC "S", and the whole Scan request will be stopped for FUNC "I". |

## PACEX036 - Object Origin Deletion

This user exit is used in a migration to synchronize objects in the origin status with PAC by creating the input data to be used for the deletion of objects at the origin status.

During the processing of the event, the objects in the origin status may be on another CPU or database and inaccessible to PAC. To synchronize the objects in the origin status with PAC, a job may be run at the location of the origin status.

This routine creates the input data to be used for the deletion at the origin status. The input data may be used as input to either SYSMAIN or NATUNLD. This exit is only invoked when the FROM-STATUS of the event is of type (T), and the TO-STATUS us either of type P, M or D .

### Parameters

| Parameter | Format | Description |
|---|---|---|
| REQUEST | A1 | Modifiable; F: first (initial) call; N: normal call; L: last call |
| OBJECT-LIB | A8 | Modifiable; name of the library in which the object resides |
| OBJECT-NAME | A32 | Modifiable; name of the object (only if N - i.e., "Normal") |
| OBJECT-TYPE | A4 | Modifiable; object type code (only if N - i.e., "Normal") |
| OBJECT-VER | P5 | Modifiable; version of the object (only if N - i.e., "Normal") |

| Parameter | Format | Description |
|---|---|---|
| OBJECT-STATUS | A32 | Modifiable; origin status of the object |
| USER-AREA | A50 | Modifiable; area for user to use |
| PAC-ERR | P5 | Not modifiable; error message number |

### Example

```
SYSMAIN DELETE,STOWED,object-name IN LIBRARY object-library
```

## PACEX037 - Predict Generation

This interface exit restricts the generation of DDMs/views or rules during Predict migrations and optionally verifies the database ID for files linked to multiple databases.

When each DDM/view to be generated is passed to this exit, the user may set the RESP code to a non-zero value to suppress the generation, and/or place a message in the MSG parameter for PAC to write into the audit report.

Additionally, in the case of files linked to multiple databases, the user may override the database ID for which the generation must be performed. PAC uses information from the generation defaults to determine which default database to use. This is only valid for files linked to multiple databases.

- If the Specification DB-ID is set to N, the database ID is not used.
- If the Specification DB-ID is set to Y and the database ID is set to blank, the Predict generation results in an error.

The user will be at End of Transaction (ET) status on entry into the routine. Control should be returned to PAC at termination of the routine.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| FUNC | A1 | Not modifiable |
| OBJECT | A32 | Not modifiable; name of a view or rule |
| DATABASE-ID | A32 | Modifiable |
| OBJ-TYPE | A15 | Not modifiable; name of the object type |
| MSG | A78 | Modifiable; error message text |
| RESP | N4 | Modifiable; return doe number. If set to not equal to 0, the DDM will not be generated. |

## PACEX038 - Event Authorization Verification

This user exit allows you to implement additional controls for event authorization before it is actually authorized. The event is invoked before and after authorization begins. Any required processing may be performed in this routine. The user will be at End Transaction (ET) status on entry into the routine.

PACEX038 is first called "before first validation". FUNCTION = 'B' at this call. The subsequent activity will be of one of four kinds, the kind depending on the value PACEX038 puts into MSG-NO.

- If MSG-NO is set at 0, then the PACEX038 specified values of the in/out fields replace the corresponding values of the event record fields and the authorisation continues as usual.
- If MSG-NO is set at 1, then the PACEX038 specified values of the in/out fields replace the corresponding values of the event record fields and the authorisation continues without the Migration Event Authorisation screen being displayed and without confirmation being required.
- If MSG-NO is set at 2, then the PACEX038 specified values of the in/out fields replace the corresponding values of the event record fields and the authorisation continues with the Migration Event Authorisation screen being displayed protected and with confirmation being required.
- If MSG-NO is set at anything else, then the authorisation is aborted. At batch authorisation the four kinds are reduced to two: MSG-NO = 0 OR 1 OR 2 or not.

If confirmation is required but does not happen, then the authorisation is aborted.

If the authorisation has not been aborted, then PACEX038 may then be called an indefinite number of times online and once in batch "after first validation". FUNCTION = 'A' at this call. If PACEX038 puts 0 into MSG-NO at this call, then the authorisation will be finished.

If PACEX038 puts anything else into MSG-NO at this call online, then the Migration Event Authorisation screen will be displayed with confirmation being required.

If PACEX038 puts anything else into MSG-NO at this call in batch, then the authorisation is aborted. At PACEX038 "after" calls PAC disregards the values PACEX038 puts into the parameter fields other than MSG-NO. The "after" calls of PACEX038 will recur online as long as the user keeps confirming and PACEX038 keeps MSG-NO at non-0.

Switch for Authorization: Alignment events (from CONTROL to CONTROL) and events of migrations from neighbour statuses to CONTROL provide a new tripositional switch - Catalogue - to be set at authorisation time. The field COPY-MOVE-INCLUDE is designated for double purposes. PACEX038 is PACEX030 enhanced.

### Possible Use

- Automatic override of parameters for certain authorizations;
- Additional audit of the security checking for authorizations;
- Set up of parameters not previously defined in the migration path.

### Parameters

| Parameter | Format | Description |
|---|---|---|
| FUNCTION | A1 | in |
| EVENT | A32 | in |
| APPLICATION | A32 | in |
| STATUS-FM | A32 | in |
| STATUS-TO | A32 | in |
| BATCH-ONLINE | A1 | in / out |
| WORKFILE-USE | A1 | in / out |
| COPY-MOVE-INCLUDE | A1 | in / out |
| REPLACE | A1 | in / out |
| JOB | A32 | in / out |
| EXP-OPTION | A1 | in / out |
| EXP-STATUS | A32 | in / out |
| APPLYMODS | A50 | in / out |
| SCHED-TIME | T | in / out |
| FTT | A32 | in / out |
| MSG-NO | P5 | out |
| MSG | A78 | out |
| USER-AREA | A64 | in / out |

## PACEX040 - Remote Node and Dataset Security Check

PACEX040 is called whenever the current user does not have logon access to the requested Entire System Server node. It allows you to specify a different user ID and/or password and the password for protected PDSs.

Using PACEX040, you can

- change the value of a USERID field and
- specify the value of a PASSWORD field.

The authorization of the resulting pair to access the node is checked by PAC.

| Parameter | Format | Description |
|---|---|---|
| FUNCTION | A1 | Input; constant "N". |
| NODE | N3 | Input; the number of the node being accessed. |
| DATA-SET-NAME | A54 | Input; the name of the dataset being accessed. |
| USERID | A8 | Input/modifiable; the current value of *INIT-USER. The ID to be used at the authorization check. |
| PASSWORD | A8 | Modifiable; the password for node access |

## MIGEX003 - File Security Check

### Function

This exit is invoked whenever PAC needs to access a Natural or Predict file to retrieve or store Natural or Predict objects. This allows the user to establish the password and cipher code to be used when PAC accesses these files.

### Remarks

Although this exit is defined as User Exit PACEX003, it is not invoked with this name; instead, exit MIGEX003 is invoked. This is because PAC comprises three subsystems loaded into libraries SYSPAC, SYSPAA and SYSTEM; and, in order to minimize the amount of maintenance, all three subsystems use exit MIGEX003 to obtain the password and cipher information needed to access the Natural/Predict system file.

Normally, when using Natural, passwords and cipher codes may be specified through the Natural parameter modules (for example, FNAT, FUSER, FDIC, NTFILE, SYSCIP, SYSPSW); however, depending on the user's security definition, this may not be adequate. MIGEX003 allows the user to provide the relevant password or cipher definitions for any and every file accessed by PAC. PAC passes the DBnr and Fnr of the file to be accessed so that the exit may return the corresponding password and cipher.

When the database and/or file number to be accessed by PAC must be translated to another database and/or file number, the exit may be used to perform the translation.

## Parameters

| Parameter | Format | Description |
|---|---|---|
| DBID | N5 | Modifiable; database ID number |
| FNR | N5 | Modifiable; file ID number |
| PSWD | A8 | Modifiable; password |
| CIPH | A8 | Modifiable; cipher |

# 6 System Applymods

This chapter describes the applymods currently used by PAC. Refer to the section Applymod Defaults chapter PAC Administrator Functions in the PAC Administration documentation for more information.

This chapter covers the following topics:



# Applymod Groups

PAC groups its system applymods as follows:

| Group | Members | Description |
|---|---|---|
| Authorizations | 1 | Disallows override of authorization parameters. |
| Migrations | 2, 4-6, 8-15, 17-19, 23, 24, 29, 30, 31 | Customize migrations. |
| Migration events | 16, 20 | Customize migration events. |
| Not assigned | 3, 7, 15, 17, 21, 22, 25-28 | Currently not assigned. |

# Applymod Functions

## Authorizations Group

| Num. | Function |
|---|---|
| 1 | Disallow override of parameters during authorization.<br>During authorization, the authorizer can override certain parameters (for example, job). This applymod deactivates that facility. |

## Migrations Group

| Num. | Function |
|---|---|
| 2 | Do not optimize the size of the cataloged object.<br>During the unload of a cataloged object, the symbol table of an object may be omitted without affecting the integrity of the object. The benefits of using the applymod are less resource usage and improved performance (during migration only).<br>Do not use this option if the cataloged object is being downloaded to a PC for use with .DBF files or with dBase III (that is, an extension of .DBF) because the field names will then not be available to Entire Connection.<br>Also do not use it if the symbol table information is required. |

| Num. | Function |
|---|---|
| 3 | Not assigned. |
| 4 | Suppress audit report.<br>This applymod may be used to suppress the creation of the audit report, thus conserving space on the ACF system file. This may be desirable for large events where the retention of the audit report is not necessary. Currently, it is not possible to purge only the audit report of an event. During migration processing, the audit report is usually written into the ESIZE before being stored on the PAC ACF system file. Because the object list is also processed in the ESIZE, a conflict in resource usage can be caused. To avoid this conflict, this applymod can be set on so that the audit report will not be written to the ESIZE. The advantage is that larger object lists may be used without incurring a NAT0886 error. The disadvantage is that the audit report will not be stored on the ACF system file. |
| 5 | Suppress global data area (GDA) Auto Expand during migration.<br>During migration processing, if PAC finds a GDA, it will automatically include (migrate) all objects that use that GDA to ensure that all objects that use it are compiled with the version being migrated, This is because a NAT0933 error may otherwise occur during execution time in the destination environment. If the user does not want PAC to perform the automatic expand, then the applymod should be specified during the actual migration. |
| 6 | Migrate only objects previously migrated.<br>During migration processing, migrate only those objects that already exist (with the same or different version number) in the assigned destination status. If an object does not already exist in the destination status, it is ignored in the migration. |
| 7 | Not assigned. |
| 8 | Force unloading of all views or local/parameter data areas during migration.<br>During migration processing, PAC unloads only VSAM views, because only VSAM views are required at execution time. All other views are used at compile time. If you want all views or local/parameter data areas to be unloaded, activate Applymod 8 for the migration. If Applymod 8 is activated, Applymod 2 will automatically be activated for the migration. If the destination status supports dynamic source programs, then Applymod 8 should be activated. |
| 9 | Expand maps for multiple languages.<br>Without this applymod, the "&" notation used by an object is normally replaced with the current value of *LANGUAGE. When this applymod is switched on, the expand will result in all valid objects that exist for languages 1-9 being included in the migration. This means that PAC systematically replaces the "&" with the values 1-9 and includes each valid occurrence for migration processing. |
| 10 | Migrate objects to a status only if they have changed.<br>During the unload step of an OUT migration, all objects in the object list are unloaded by default; this ensures that the objects are loaded as a unit. It is possible that the environment does not need to be refreshed. When this applymod is switched on, only new versions of objects will be migrated; unchanged versions of the objects will be ignored. Only valid when the to-staus is either test, production, or maintenance. |
| 11 | Check runtime subordinates.<br>Check whether all runtime subordinates used by each PAC object in the Object List in a migration from development or maintenance are currently under PAC control. If any subordinate is not found and this applymod is set, PAC issues the PAC7028 message and the migration event is aborted. |
| 12 | Check object compile date-time stamp.<br>Retain those objects with the most recent compile date-time stamp during migrations from test or production statuses. PAC always migrates the specified version of the object to the specified status. |

| Num. | Function |
|---|---|
| | If this applymod is on, PAC checks the compile date-time stamps of the version being migrated and the version being replaced. If the object being replaced has a more recent compile date-time, it is not replaced. |
| 13 | Activate a recursive expand of objects when expanding.<br>By default, PAC expands only one level of objects in the object list; it does not subsequently expand the objects that are added to the object list by the first expansion. If Applymod 13 is turned on, PAC expands the object list recursively; when objects are added to the list by the first expansion, the new objects are then expanded, and so on. |
| 14 | Reject objects with compilation errors.<br>During the migration from a development or maintenance status type into PAC, new versions of objects are created regardless of the compile results of the object. When this applymod is active, no new version will be created for the object if it does not compile successfully. If the Move option is in effect for the migration event, then this applymod is ignored. |
| 15 | Unload Predict file without associated objects. Enables the user to unload a Predict file out of PAC without its linked objects, such as Userviews, Databases etc. |
| 17 | Load Predict file without Adabas attributes. Enables the user to load a Predict File into PAC ignoring its Adabas attributes. |
| 18 | Suppress building the object list for incorporation.<br>With the incorporation migration, PAC requires the original object list to be a single entry with or without the range notation when including objects from a work file. During migration processing, PAC updates this list with the expanded list of individual objects actually incorporated. If this applymod is active, the object list is not expanded. This applymod may be required with large migrations when the total list of objects exceeds the ESIZE. |
| 19 | Force the migration of saved copy code, programs, and maps.<br>By default, PAC migrates only cataloged objects (except text objects) to environments other than development or maintenance. If Applymod 19 is turned on, PAC always migrates both source and cataloged objects. |
| 22 | Not assigned. |
| 23 | Allow objects in any status to be selected for archiving, purging, or retiring.<br>Use for archiving objects, object maintenance (purge function), and retiring objects. 1) Allow objects in any status to be selected for archiving. If this applymod is not active, only objects in Control status can be selected for archiving. Objects are removed from the status and placed in the Archive status. 2) Allow objects in any status (except Archive) to be selected for purging. If this applymod is not active, only objects in Control status can be purged. Objects in Archive can be separately purged. 3) Allow objects in any status to be selected for retiring. If this applymod is not active, only objects in Control and Archive statuses can be retired. |
| 24 | Suppress purging of cataloged objects with Move option.<br>Allow cataloged objects in a maintenance status type not to be purged with the Move option for migrations into PAC from maintenance or development status types. |
| 25 | Not assigned. |
| 26 | Not assigned. |
| 27 | Not assigned. |
| 28 | Not assigned. |

| Num. | Function |
|---|---|
| 29 | Compare object save date and event authorization date.<br>For each object on the Object List during a migration from development or maintenance, PAC compares the date the object was saved with the Event's authorization date. If the object save date is later than the authorization date and this applymod is set, PAC terminates the migration event. |
| 30 | Control message scrolling.<br>This applymod toggles the automatic scrolling of PAC messages during an online migration. |
| 31 | Do not migrate cataloged objects to a status.<br>No: Migrate the loadable part of a Natural object whenever the object is migrated.<br>Yes: Do not migrate the loadable part of a Natural object to a status. |

## Migration Events Group

| Num. | Function |
|---|---|
| 16 | Update object list with the highest version object found.<br>During Expand, PAC ignores objects resulting from the Expand process that already exist in the object list. When this applymod is active, PAC compares the versions of the two objects and updates the object list with the highest version found. |
| 20 | Retain the job name specified in the source event when an event is being copied.<br>During a Copy Event function, this applymod instructs PAC to retain the job name specified in the source event rather than use the job name specified in the migration path.<br>During group event submission, this applymod allows PAC to submit the group event even when one or more of its member events is not in Authorized state. |

# 7 Object Type Codes

The following tables list and describes the object type codes used in PAC.

**Note:** Non-Natural types are site-specific and defined by the PAC administrator.

This chapter covers the following topics:

- **Natural Object Type Codes**
- **Predict Object Type Codes**

## Natural Object Type Codes

| | |
|---|---|
| A | Parameter Data |
| C | Copycode |
| E | Error message |
| G | Global Data |
| H | Helproutine |
| L | Local Data |
| M | Map |
| N | Subprogram |
| P | Program |
| R | Rule |
| S | Subroutine |
| T | Text |
| V | View |
| Y | Expert Model |
| Z | Recording |
| 3 | Dialog |
| 4 | Class |
| 5 | Processor |
| 7 | Function |
| * | All object types |

**Note:** Error messages, DDMs and rules may, in some cases, be listed together with Natural objects for migrations of applications other than Predict. In these cases, their type denotations are "E" (for error messages), "View" (for DDMs) and "Rule". Some screens take "V" for DDMs and "R" for rules.

## Predict Object Type Codes

| Type Code | Description |
|---|---|
| DA | Predict database |
| DS | Predict dataspace |
| FIA (FILE) | Predict Adabas file |
| FIC | Other Predict file |
| FIU (VIEW) | Predict userview |
| KY | Predict keyword |
| NW | Predict network |
| PA | Predict packagelist |
| PR | Predict program |
| RE | Relationship |
| RULE | Rule (generated from verification) |
| ST | Predict storagespace |
| SY | Predict system |
| US | Predict user |
| VE | Predict verification rule |
| VM | Virtual machine |
| Y | Expert model |

# II Application Programming Interface

PAC's Application Programming Interface (API) facilities are Natural subprograms invoked using CALLNAT statements to access PAC directly from within a user-written program without invoking PAC with MENU, ADMIN, or SYSPAC commands, see **Invoking the APIs**.

The APIs provide a wide range of flexibility, making it easier for you to customize the interface with PAC to meet your site requirements.

PAC provides the following APIs:

| API Routine | PAC Entity | PAC Function |
|---|---|---|
| **APINAPP** | Application | Add, Display, Modify |
| **APINAPPL** | Application | Add, Display, Modify |
| **APINAUTH** | Event Authorization | Add, Display, Purge |
| **APINCCLG** | Change Logs | Display, Purge |
| **APINDSC** | Event, Maintenance Request, Job | Display, Modify, Purge |
| **APINENTS** | Common, PAC and PAA objects, PAC JCL texts (jobs) and events | List |
| **APINJOB** | Job | Display |
| **APINLCKS** | Locked Data | Add, Display, Modify |
| **APINLNK** | Application Status Link | Add, Display, Modify |
| **APINLNKS** | PAC application - status links | List |
| **APINLSRC** | Object Version Natural Source | Load (read into Natural work area) |
| **APINMEOL** | Migration Event Locked Object | Display |
| **APINMGEV** | Migration Event | Add, Display, Modify, Submit, Purge |
| **APINMR** | Maintenance Request | Add, Display, Modify, Purge |
| **APINNSRC** | Objects, Used Lists | Display |
| **APINOBJV** | Object Version | Display |
| **APINOBLL** | Migration List | Add, Display, Modify |
| **APINOBLS** | Object List | Add, Display, Purge |

| API Routine | PAC Entity | PAC Function |
|---|---|---|
| **APINOSLS** | PAC object - status links | List |
| **APINPATH** | Migration Path | Add, Display, Modify |
| **APINPRF** | User Profile | Add, Modify, Retrieve |
| **APINPRFS** | User Profile | Select |
| **APINSEL** | Application<br>Application Status Link<br>Maintenance Request<br>Migration Event<br>Migration Path<br>Job<br>Object Version<br>Status | Select |
| **APINSELE** | Event handling | Select |
| **APINSELO** | Object Status Link | Display |
| **APINSEVT** | Event handling | Submit |
| **APINSTAT** | Status | Add, Display, Modify |
| **APINSTFF** | Objects, Used Lists | Display |
| **APINSTT** | Status | Add, Display, Modify |
| **APINTEXL** | Migration List | Add, Display, Modify |
| **APINTEXT** | Event, Maintenance Request, Job | Display |

Library SYSPACUS contains example programs for each of the individual APIs. These programs can be tailored to meet the specific needs of your site.

Invoke UGNXMAIN to access the API example programs.

How to invoke PAC APIs is described in **Invoking the APIs**.

# 8 Invoking the APIs

During execution, the API routines use PAC subroutines residing in the SYSPAC and SYSTEM libraries.

1. If the application that calls the PAC API subroutines does not reside in the SYSPAC library, define the SYSPAC library as steplib to the Natural session.
2. Use the following syntax to invoke a PAC API:

```
CALLNAT  'APINxxxx'  UGNLPARM  UxxLyyyy
```

   where

   - the first parameter (supplied as local data area UGNLPARM) has a fixed format; and
   - the second parameter (supplied as UxxLyyyy) has a variable format, depending on the particular API call. The variable format for each API is detailed in the following API descriptions.

This chapter covers the following topic:

- **Local Data Area UGNLPARM**

## Local Data Area UGNLPARM

The first parameter of all PAC APIs is the following structure API-PARM. Its definition is provided in the local data area UGNLPARM:

| Field Name | Format | Description |
|---|---|---|
| API-PARM | - | - |
| API-FUNC (R) | A2 | (optional) Identifies the code of the subfunction for the request. |
| API-MAIN-FUNC | A1 (I) | The code of the function to execute on the entity. |
| API-SUB-FUNC | A1 (I) | (optional) Identifies the code of the subfunction for the request. |
| API-MSG-NO | P5 (O) | PAC error message number. |
| API-MSG | A78 (O) | Detailed message of any error occurred during processing. |
| API-USER-AREA | A50 (I/O) | Area used by additional routines or exits invoked during the processing of the API request. |

# 9 APINAPP

API Routine: APINAPP

Parameters: UGNLPARM, UPILAPP

This API lets obtain data from a PAC application, create a new application, and modify an existing one.

APINAPP can be invoked with:

```
CALLNAT 'APINAPP' API-PARM API-APP-PARM
```

A definition of API-PARM is provided in LDA UGNLPARM in library SYSPACUS.

Set API-PARM.API-MAIN-FUNC to:

A To add a Natural application.

B To add a Predict application.

D To obtain data from an application.

M To modify an application.

This chapter covers the following topics:

- **Local Data Area UPILAPP**
- **API Return Codes**

## Local Data Area UPILAPP

A definition of API-APP-PARM is provided in LDA UPILAPP in library SYSPACUS.

| Field Name | Format | Description |
|---|---|---|
| API-APP-PARM | | |
| APPL-TYPE | A1 | out |
| APPLICATION | A32 | in |
| PREFIX | A3 | in/out |
| LEVEL | A10 | in/out |
| GEN-NUMBER | P5 | in/out |
| TITLE | A60 | in/out |
| MCG | A4 | in/out |
| PRD-APPL | A32 | in/out |

| Field Name | Format | Description |
|---|---|---|
| VIEW-SECURITY | A1 | in/out |
| STEPLIB-CNT | N3 | out |
| STEPLIB | A32 (1:8) | in/out |
| FRG-TYPE-CNT | N3 | out |
| FRG-TYPE | A4 (1:100) | in/out |
| MIN-CURR-VERS | P3 | in/out |
| CURR-RETENTION-PERIOD | P3 | in/out |
| MIN-ARCH-VERS | P3 | in/out |
| ARCH-RETENTION-PERIOD | P3 | in/out |
| LOG-PURGES | A1 | in/out |
| PRD-CASE-USAGE | A1 | in/out |
| DB2-SUPPORT | A1 | in/out |
| LIB-DEV | A8 | in/out |
| LIB-INC | A8 | in/out |
| LIB-MAINT | A8 | in/out |
| LIB-TEST | A8 | in/out |
| ADD-TIME | T | out |
| ADD-BY | A8 | out |
| ADD-TID | A8 | out |
| MOD-TIME | T | out |
| MOD-BY | A8 | out |
| MOD-TID | A8 | out |

Whenever APINAPP is called, API-APP-PARM.APPLICATION should carry the name of the application.

If [API-PARM.API-MAIN-FUNC] = 'D' and the application is found, APINAPP returns in the remaining parameter fields the application data, ignoring any values put by the user in the "in/out" fields.

[API-APP-PARM.STEPLIB-CNT] is the number of the application's step applications. Their names are returned in API-APP-PARM.STEPLIB(1:[API-APP-PARM.STEPLIB-CNT]).

[API-APP-PARM.FRG-TYPE-CNT] is the number of the foreign object types associated with the application. The codes of the types are returned in API-APP-PARM.FRG-TYPE(1:[API-APP-PARM.FRG-TYPE-CNT]).

If [API-PARM.API-MAIN-FUNC] = 'A' and the application is not found, APINAPP tries to add a NATURAL application with the characteristics specified in the "in/out" fields.

If [API-PARM.API-MAIN-FUNC] = 'B' and the application is not found, APINAPP tries to add a PREDICT application with the characteristics specified in the "in/out" fields.

If [API-PARM.API-MAIN-FUNC] = 'M' and the application is found, APINAPP tries to modify the application endowing it with the characteristics specified in the "in/out" fields.

The meaningful values of the "in/out" fields are enumerated in the following table:

| Field Name | Format | Possible Values |
|---|---|---|
| PREFIX | A3 | any |
| LEVEL | A10 | any |
| GEN-NUMBER | P5 | any non-negative |
| TITLE | A60 | any |
| MCG | A4 | a sequence of I, M, N, O, W, Y |
| PRD-APPL | A32 | the name of a Predict application or blank |
| VIEW-SECURITY | A1 | A, D, U |
| STEPLIB | A32 (1:8) | the names of Natural applications or blank |
| FRG-TYPE | A4 (1:100) | codes of foreign object types or blank |
| MIN-CURR-VERS | P3 | any non-negative |
| CURR-RETENTION-PERIOD | P3 | any non-negative |
| MIN-ARCH-VERS | P3 | any non-negative |
| ARCH-RETENTION-PERIOD | P3 | any non-negative |
| LOG-PURGES | A1 | N, Y |
| PRD-CASE-USAGE | A1 | N, Y |
| DB2-SUPPORT | A1 | N, Y |
| LIB-DEV | A8 | any |
| LIB-INC | A8 | any |
| LIB-MAINT | A8 | any |
| LIB-TEST | A8 | any |

With any of the four meaningful values of API-PARM.API-MAIN-FUNC the significance of the values of the API-APP-PARM fields is the same as that of their menu system counterparts.

At application addition or modification user-specified values of the "in/out" fields may be translated.

Embedded spaces are replaced with '_' in user-specified values of PREFIX and LEVEL.

A meaningless value of PRD-APPL specified for the addition or modification of a Natural application causes the function to be abandoned. Other meaningless values of the "in/out" fields are treated as follows:

| Field Name | Format | Possible Values |
|---|---|---|
| GEN-NUMBER | P5 | A negative is translated to 0. |
| MCG | A4 | Meaningless characters are skipped. |
| VIEW-SECURITY | A1 | Meaningless character is translated to 'U'. |
| STEPLIB | A32 (1:8) | Strings that are not Natural application names are skipped. |
| FRG-TYPE | A4 (1:100) | Strings that are not codes of foreign object types are skipped |
| MIN-CURR-VERS | P3 | A negative is is translated to 0. |
| CURR-RETENTION-PERIOD | P3 | A negative is is translated to 0. |
| MIN-ARCH-VERS | P3 | A negative is is translated to 0. |
| ARCH-RETENTION-PERIOD | P3 | A negative is is translated to 0. |
| LOG-PURGES | A1 | A meaningless character is translated to 'N'. |
| PRD-CASE-USAGE | A1 | A meaningless character is translated to 'N'. |
| DB2-SUPPORT | A1 | A meaningless character is translated to 'N'. |

## API Return Codes

APINAPP returns the following [API-PARM.API-MSG-NO]:

1011 if an [API-PARM.API-MAIN-FUNC] distinct from 'A', 'B', 'D', and 'M' has been specified.

7201 if an ACF control record has been found defective.

7205 if the "initialisation" has failed, ie, logical files 210 and 211 have not been found set at a pair of matching ACF and PCF.

7250 if the application has been found.

7251 if an application has not been found.

7253 if the application has been added.

7254 if the application has been modified.

7257 if the application has not been added.

7259 if the application has not been modified.

7541 if the application (to be modified) is locked.

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.

# 10 Application API

API Routine: APINAPPL

Parameters: UGNLPARM, UAPLAPPL

The Application API allows users to add, modify or display (retrieve) information about a particular application. The following functions are valid:

A Add application.

D Display application (Retrieve).

M Modify an existing application.

This chapter covers the following topics:



## Local Data Area UAPLAPPL

The fields to be passed to the API as the second parameter are in the supplied local data area UPILAPP. The following is a detailed description of these fields:

| Field Name | Format | Description | |
|---|---|---|---|
| APPLICATION | A32 | Name of the application. You may use application name or alias for the Display and Modify functions. | |
| TITLE | A60 | Title of the application. | |
| PREFIX | A3 | Prefix of the application. | |
| LEVEL | A10 | Level of the application. | |
| GEN-NUMBER | P5 | Number to be used during event name generation. | |
| PRD-XREF-FORCE | A1 | Force Predict documentation. Valid options are: | |
| | | Y | Predict documentation entries must be present before a Natural program can be migrated into PAC. |
| | | N | Default. |
| PCA-USAGE | A1 | Predict Case usage. Valid options are: | |
| | | Y | Natural objects for this application have been composed by Predict Case. |
| | | N | Default. |
| DEF-NAT-LIB-DEV | A8 | Name of the default Development library. | |

| Field Name | Format | Description | |
|---|---|---|---|
| DEF-NAT-LIB-INC | A8 | Name of the default Incorporation library. | |
| DEF-NAT-LIB-MAINT | A8 | Name of the default Maintenance library. | |
| #FILLER-PROD | A8 | Not used. | |
| DEF-NAT-LIB-TEST | A8 | Name of the default Test library. | |
| MCG | A4 | The machine code generator MCG option. Valid options are: | |
| | | blank | MCG is not used. |
| | | N | MCG OFF |
| | | Y | MCG ON |
| | | I | INDX is to be activated. |
| | | O | OVFLW is to be activated. |
| | | M | MIX is to be activated. |
| FILE-SECURITY | A1 | Default DDM use mode. Valid options are: | |
| | | A | Reading access only. |
| | | D | Disallowed. |
| | | U | Update. |
| MIN-CURR-VERS | P3 | Minimum current versions. | |
| CURR-RETENTION-PERIOD | P3 | Current retention period. | |
| MIN-ARCH-VERS | P3 | Minimum archive versions. | |
| ARCH-RETENTION-PERIOD | P3 | Archive retention period. | |
| LOG-PURGES | A1 | Log all object purges. Valid options are: | |
| | | Y | All purges through the Purge Object Version function are to be logged. |
| | | N | Default. |
| MOD-TIME | T | Date and time application was last modified. | |
| MOD-BY | A8 | User that last modified application. | |
| MOD-TID | A8 | Terminal that last modified application. | |
| ADD-TIME | T | Date and time application was added. | |
| ADD-BY | A8 | User that added application. | |
| ADD-TID | A8 | Terminal that added application. | |
| DB2-SUPPORT | A1 | Access to DB2 files. Valid options are: | |
| | | Y | Yes. |
| | | N | No. |
| APPL-TYPE-CODE | A1 | Type of the application. Valid options are: | |
| | | D | Predict |
| | | N | Natural |
| APPL-SYSDIC-NAME | A32 | Name of the Predict application carrying the DDMs for application [APPLICATION]. | |

| Field Name | Format | Description |
|---|---|---|
| COMPILE-STEPLIBS-1-4 | A32 (1:4) | Names of default step applications 1 to 4. |
| COMPILE-STEPLIBS-5-8 | A32 (1:4) | Names of default step applications 5 to 8. |

## Application API Return Codes

7250 Application exists.

7251 Application not found.

7253 Application added.

7254 Application has been modified.

# 11 Event Authorization API

API Routine: APINAUTH

Parameters: UGNLPARM, UMELAUTH

The Event Authorization API allows users to add, display, or purge information about a particular migration event authorization. The following functions are valid:

A Authorize a migration event.

D Display authorization information for a specific migration event.

P De-authorize a migration event.

This chapter covers the following topics:

- **Local Data Area UMELAUTH**
- **Event Authorization API Return Codes**

## Local Data Area UMELAUTH

The fields to be passed to the API as the second parameter are in the supplied local data area UMELAUTH. The following is a detailed description of these fields:

| Field Name | Format | Description | |
|---|---|---|---|
| EVENT | A32 | Name of the event. | |
| FM-LIBRARY | A8 | Output - From library name. | |
| FM-FUSER-DBID | N5 | Output - Database of the Natural system file where the From library is located. | |
| FM-FUSER-FNR | N5 | Output - File of the Natural system file where the From library is located. | |
| FM-FDIC-DBID | N5 | Output - Predict database for the From library. | |
| FM-FDIC-FNR | N5 | Output - Predict file for the From library. | |
| TO-LIBRARY | A8 | Output - To library name. | |
| TO-FUSER-DBID | N5 | Output - Database of the Natural system file where the To library is located. | |
| TO-FUSER-FNR | N5 | Output - File of the Natural system file where the To library is located. | |
| TO-FDIC-DBID | N5 | Output - Predict database for the To library. | |
| TO-FDIC-FNR | N5 | Output - Predict file for the To library. | |
| JOB | A32 | Name of the job that is to perform the migration event. | |
| EXPAND-OPT | A1 | Auto Expansion option. Valid options are: | |
| | | C | Include all subordinates |

| Field Name | Format | Description | |
|---|---|---|---|
| | | E | Include all subordinate and referenced objects |
| | | R | Include referenced objects |
| | | U | Include used-by objects |
| | | N | Do not Auto Expand |
| EXPAND-STATUS | A32 | Name of the status to be used during object list expand. | |
| BATCH-ONLINE | A1 | Run migration event in batch or online. Valid options are: | |
| | | B | Migration event is to be run in batch. |
| | | O | Migration event is to be run online. |
| WORKFILE-USAGE | A1 | Work file usage. Valid options are | |
| | | Y | (batch migrations only) Work file is to be used during the processing of migration events. |
| | | N | Objects to be migrated directly to the target location without using a work file. |
| COPY-MOVE-INCLUDE | A1 | Copy, move or include objects. | |
| | | C | Objects are to be copied, retaining a copy in the source location. |
| | | M | Objects are to be moved from the source location with no copy retained. `Move' is only valid when the origin status is a development, test, or maintenance status. |
| | | I | Objects are to be included from a work file. `Include' is only valid for batch migrations when the origin status is a development or incorporation status. |
| REPLACE | A1 | Replace objects at target destination (Y/N). Valid for development and maintenance destination statuses only. | |
| SCHED-DATE | (T) | The earliest date and time this event may be submitted. | |
| APPLYMODS | A50 | Applymods(s) to be switched on during processing of the migration event. | |
| TRANSLATION-TABLE | A32 | Name of the file translation table (FTT) to be used during processing of the migration event. | |
| CAT-FLAG | A1 | Alignment events (from CONTROL to CONTROL) and events of migrations from neighbour statuses to CONTROL provide a tripositional switch to be set at authorization time. Valid options are: | |
| | | D | Default value. Natural objects will be catalogued only if they have been previously catalogued. |
| | | N | Natural objects will not be catalogued. |

| Field Name | Format | Description | |
|---|---|---|---|
| | | Y | Natural objects will be cataloqued regardless of how they have been immigrated into PAC. |
| GENERATION-DEFAULTS | A8 | Predict Generation Defaults to be used during the migration. | |
| AUTH-TIME | T | Date and time event was authorized. | |
| AUTH-BY | A8 | User that authorized event. | |
| AUTH-TID | A8 | Terminal that authorized event. | |

## Event Authorization API Return Codes

7251 Migration Event not found.

7253 Migration Event authorization added.

# 12 APINCCLG

This API lets the user retrieve and purge a change control log. It also allows the user to purge an open change control log.

APINCCLG can be invoked with:

```
CALLNAT 'APINCCLG' API-PARM #API-CCLG-PARM
```

D to retrieve a change control log.

P to purge an open change control log.

**Important:** Reset the field SEL-DATE programmatically before the call is made to APINCCLG.

This chapter covers the following topics:

- **Local Data Area UCCLCCLG**
- **Application API Return Codes**

## Local Data Area UCCLCCLG

A definition of #API-CCLG-PARM is provided in LDA UCCLCCLG.

| Field Name | Format | I/O |
|---|---|---|
| #API-CCLG-PARM | | |
| #API-CCLG-PARM-1 | A250 | |
| #API-CCLG-PARM-1 | | |
| APPLICATION | A32 | in |
| OBJECT | A8 | in |
| LIBRARY | A8 | in |
| DBID | N5 | in |
| FNR | N5 | in |
| VERSION | N4 | in |
| SEL-DATE | T | in |
| REQUEST-ID | A20 | out |
| STATUS | A32 | out |
| OUT-VERSION | N4 | out |
| OUT-LIBRARY | A8 | out |
| OUT-DBID | N5 | out |
| OUT-FNR | N5 | out |

| Field Name | Format | I/O |
|---|---|---|
| OUT-USERID | A8 | out |
| OUT-TIME | T | out |
| OUT-TID | A8 | out |
| IN-VERSION | N4 | out |
| IN-LIBRARY | A8 | out |
| IN-DBID | N5 | out |
| IN-FNR | N5 | out |
| IN-USERID | A8 | out |
| IN-TIME | T | out |
| IN-TID | A8 | out |

When APINCCLG is called:

- #API-CCLG-PARM.APPLICATION should carry the name of the change control log's application.
- #API-CCLG-PARM.OBJECT should carry the name of the object.
- #API-CCLG-PARM.LIBRARY should carry the library name of the NATURAL location of the maintenance deployment.
- #API-CCLG-PARM.DBID should carry the database number of the NATURAL location of the maintenance deployment.
- #API-CCLG-PARM.FNR should carry the file number of the NATURAL location of the maintenance deployment.
- #API-CCLG-PARM.VERSION should carry the object's version number.

If [API-PARM.API-MAIN-FUNC] = 'D' and the change control log is found then APINCCLG returns in the remaining parameter fields the change control log data including the name of the maintenance status and the name of the maintenance request (if one is associated with the log).

If [API-PARM.API-MAIN-FUNC] = 'P' and the change control log is found then APINCCLG purges it even if it is open.

## Application API Return Codes

APINCCLG returns the following [API-PARM.API-MSG-NO]:

1011 if an [API-PARM.API-MAIN-FUNC] distinct from both 'D' and 'P' has been specified,

7251 if the application is unknown or the change control log has not been found,

7255 if the change control log has been retrieved,

7256 if the change control log has been purged.

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.

# 13 APINDSC

This API lets the user:

- obtain a chunk of
  - the description of an event,
  - the description of a maintenance request,
  - a JCL text,
  - the audit report of an event,
  - the audit report of a finalized archiving event,
- append a chunk to
  - the description of an event,
  - the description of a maintenance request,
  - a JCL text,
  - the audit report of an event,
  - the audit report of a finalized archiving event,
- delete
  - the description of an event,
  - the description of a maintenance request,
  - a JCL text,the audit report of an event,
  - the audit report of a finalized archiving event.

APINDSC can be invoked with:

```
CALLNAT 'APINDSC' API-PARM API-DSC-PARM
```

Choose a value for API-PARM.API-MAIN-FUNC from the following table:

| | a description | a JCL text | an audit report |
|---|---|---|---|
| to obtain a chunk of | D | J | R |
| to append a chunk to | E | K | S |
| to delete | F | L | T |

Set API-PARM.API-SUB-FUNC at:

E To handle an event.

H To handle a finalized archiving event.

M To handle a maintenance request.

J To handle a JCL text.

This chapter covers the following topics:

## Local Data Area UGNLDSC

A definition of API-DSC-PARM is provided in LDA UGNLDSC.

| Field Name | Format | I/O |
|---|---|---|
| API-DSC-PARM | | |
| OBJECT-NAME | A32 | in |
| LINE-CNT | P5 | in/out |
| TEXT-RECORD | A80 (1:60) [1*] | in/out |
| API-PAC-AREA | B20 | in/out |

[1*]: This is the proposed default value. Any value equal or greater 1 can be defined by the user.

When APINDSC is called:

- API-DCS-PARM.OBJECT-NAME should carry the name of the entity (event, finalized archiving event, maintenance request, or JCL text).
- If [API-PARM.API-MAIN-FUNC] is:
  - D, J, or R
    then [API-DSC-PARM.LINE-CNT] is made 0, if negative; it is made 60, if greater than 60. It is then interpreted as the number of lines (the size of the chunk) to be returned. In particular, no lines will be returned, if LINE-CNT = 0.
  - E, K, or S
    then [API-DSC-PARM.LINE-CNT] is made 0, if negative; it is made 60, if greater than 60. It is then interpreted as the number of lines (the size of the chunk) to be appended. In particular, an empty chunk will be appended, if LINE-CNT = 0.
  - F, L, or T
    then API-DSC-PARM.LINE-CNT is not used by APINDSC.
- If [API-PARM.API-MAIN-FUNC] is:
  - D, J, or R
    then API-DSC-PARM.TEXT-RECORD will carry the returned chunk.
  - E, K, or S
    then the chunk inAPI-DSC-PARM.TEXT-RECORD(1 : [API-DSC-PARM.LINE-CNT]) will be appended.
  - F, L, or T
    then API-DSC-PARM.TEXT-RECORD is not used by APINDSC.

- Normally, API-DSC-PARM.PAC-AREA should be reset when the first chunk of the description, text, or report is wanted or is to be appended. API-DSC-PARM.PAC-AREA should be left untouched for a next call to obtain or append an immediately following chunk of the same description, text, or report.
- If [API-PARM.API-MAIN-FUNC] is D, J, or R then APINDSC returns in API-TEXT-PARM.LINE-CNT the number of lines of the returned chunk. It may be less than the adjusted original value, if the end of the description, text, or report has been reached.

## Application API Return Codes

APINDSC returns the following [API-PARM.API-MSG-NO]:

0000 if the entity has been found and a non-final chunk is being returned, a chunk has been appended, or the description, text, or report has been deleted,

1011 if the specified ([API-PARM.API-MAIN-FUNC],[API-PARM.API-SUB-FUNC]) is meaningless,

7251 if the entity has not been found,

9999 if the entity has been found and a final chunk is being returned.

# 14 APINENTS

This API lets obtain a chunk of a list of entities. They are of four kinds:

- PAA objects,
- PAC objects,
- PAC meta entities - JCL texts, migrations lists and audit reports,
- common objects

PAA and PAC objects are classified into four groups:

- DDMs,
- error messages,
- foreign objects,
- Natural objects.

PAC meta entities are classified into two groups:

- JCL texts,
- migration events.

Common objects are classified into three groups:

- DDMs,
- error messages,
- Natural objects.

APINENTS can be invoked with:

```
CALLNAT 'APINENTS' API-PARM API-ENTS-PARM
```

To obtain a chunk, set API-PARM.API-MAIN-FUNC at 'S'. API-PARM.API-SUB-FUNC is not used by APINENTS.

This chapter covers the following topics:

- **Local Data Area UPILENTS**
- **API Return Codes**

## Local Data Area UPILENTS

A definition of API-ENTS-PARM is provided in LDA UPILENTS.

| Field Name | Format | Description |
|---|---|---|
| API-ENTS-PARM | | |
| KIND | A1 | in |
| DBNR | B2 | in |
| FNR | B2 | in |
| TYPE | A18 | in |
| TYPE1 | A1 | in |
| TYPE2 | A16 | in |
| TYPE3 | A1 | in |
| E-BYTE | A1 | in |
| LIBRARY | A8 | in |
| COMPARTMENT | B2 | in/out |
| APPLICATION | A32 | in |
| LOC-LIBRARY | A8 | in |
| LOC-DBNR | B2 | in |
| LOC-FNR | B2 | in |
| LOC-ESY-NODE | N5 | in |
| LOC-PDS | A54 | in |
| LOC-VOL-TYPE | A6 | in |
| NAME | A33 | in |
| C-VERSION-FM | N5 | in |
| C-VERSION-TO | N5 | in |
| A-VERSION-FM | N5 | in |
| A-VERSION-TO | N5 | in |
| DDM-DBNR-COND-K | A1 | in |
| DDM-DBNR-COND-M | A5 | in |
| DDM-DBNR-COND-L | N5 | in |
| DDM-DBNR-COND-U | N5 | in |
| DDM-FNR-COND-K | A1 | in |
| DDM-FNR-COND-M | A5 | in |
| DDM-FNR-COND-L | N5 | in |
| DDM-FNR-COND-U | N5 | in |

| Field Name | Format | Description |
|---|---|---|
| TIME-FM | T | in |
| TIME-TO | T | in |
| USER | A9 | in |
| STATUS | A32 | in |
| EVENT | A33 | in |
| CURRENT-ON | T | in |
| STATUS-FM | A33 | in |
| STATUS-TO | A33 | in |
| UAPPLICATION | A32 | in |
| JOB-FM | N10 | in |
| JOB-TO | N10 | in |
| STATE | A1 | in |
| LINE-CNT | N3 | out |
| LINE-TOT | N3 | in/out |
| LINE | A250 (1:60) | out |
| API-PAC-AREA | A160 | in/out |

Whenever APINENTS is called, [API-ENTS-PARM.KIND] should be:

A for a list of PAA objects,

C for a list of PAC objects,

E for a list of PAC meta entities, or

M for a list of common objects.

[API-ENTS-PARM.DBNR] and [API-ENTS-PARM.FNR] must specify an FPAA, an ACF, an ACF, or an FUSER (FNAT) or FDIC respectively.

[API-ENTS-PARM.TYPE1] should be:

D for DDM,

E for error message,

F for foreign,

J JCL text,

M migration event, or

N Natural.

The six possible settings of API-ENTS-PARM.TYPE1 can be usefully combined with the following values of API-ENTS-PARM.TYPE3:

| TYPE1 | TYPE3 | |
|---|---|---|
| D | ' ' | for DDMs themselves, |
| | B | for the USED BY lists (of DDMs in a PAC compartment). |
| E | ' ' | for error messages themselves, |
| | B | for the USED BY lists (of DDMs in a PAC compartment). |
| F | L | for loadables, |
| | S | for sources. |
| J | ' ' | . |
| M | A | for the audit reports, |
| | I | for the migration lists. |
| N | B | for the USED BY lists (of Natural objects in a PAC compartment), |
| | L | for loadables, |
| | S | for sources, |
| | U | for the USED lists (of Natural objects in a PAC compartment). |

If [API-ENTS-PARM.TYPE1] = 'F', then API-ENTS-PARM.TYPF2 should carry the PAC foreign type code of the objects to be selected with the leading '3' pared off (eg, '3JCL' would turn into 'JCL').

If [API-ENTS-PARM.TYPE1] = 'N', then API-ENTS-PARM.TYPF2 may carry a sequence of one character external codes of NATURAL types (eg, 'N' for subprogram or 'P' for program).

If [API-ENTS-PARM.TYPE1] = 'E', then API-ENTS-PARM.E-BYTE may carry a one character language code; this will restrict the selection to long error messages in the language specified.

If [API-ENTS-PARM.KIND] = 'M' and ([API-ENTS-PARM.TYPE1] = 'N' or [API-ENTS-PARM.TYPE1] = 'E'), then API-ENTS-PARM.LIBRARY should carry the name of the library whose objects are to be listed.

API-ENTS-PARM.COMPARTMENT should be reset at the first call for a list of PAC objects or events and may be left alone at the subsequent calls for the same list.

If [API-ENTS-PARM.KIND] = 'C' then API-ENTS-PARM.APPLICATION should carry the name of the PAC (ACF) compartment whose objects are to be listed. If [API-ENTS-PARM.KIND] = 'A' or ([API-ENTS-PARM.KIND] = 'E' and [API-ENTS-PARM.TYPE1] = 'M') then API-ENTS-PARM.APPLICATION may carry the name of a PAC compartment.

If [API-ENTS-PARM.KIND] = 'A', then the five LOC-... fields should carry the specification of the PAA location whose objects are to be listed.

API-ENTS-PARM.NAME should carry the specification of a range into which the names of the entities to be selected must fit.

If [API-ENTS-PARM.KIND] = 'A' or 'C', then API-ENTS-PARM.C-VERSION-FM and API-ENTS-PARM.C-VERSION-TO may carry the specification of an interval into which the PAC version numbers of the objects to be selected must fit. [API-ENTS-PARM.C-VERSION-TO] = 0 does not impose an upper bound on the PAC version numbers of the objects being selected.

If [API-ENTS-PARM.KIND] = 'A', then API-ENTS-PARM.A-VERSION-FM and API-ENTS-PARM.A-VERSION-TO may carry the specification of an interval into which the PAA version numbers of the objects to be selected must fit. [API-ENTS-PARM.A-VERSION-TO] = 0 does not impose an upper bound on the PAA version numbers of the objects being selected.

If [API-ENTS-PARM.TYPE1] = 'D', then the eight DDM-... fields may carry the specification of conditions imposed on the data base and file numbers of the files whose DDMs are to be listed. A condition is imposed on such a number if the corresponding ...COND-K field carries 'B' (between), 'G' (greater or equal), 'L' (less or equal), 'M' (mask only), 'O' (strictly outside). In any of these cases the number must fit the mask specified in the corresponding ...COND-M field, decimal digits and '.', which will match any digit in the number, being the characters acceptable in a mask. If [...COND-K] = 'M', then the mask is the only condition imposed on the number. In the other cases the number must additionally fit into or, with 'O', outside the specified interval. The lower bound should be specified in ...COND-L and the upper bound in ...COND-U. Thus [...COND-K] = 'L' makes [...COND-U] relevant, and [...COND-K] = 'G' makes [...COND-L] relevant.

If [API-ENTS-PARM.KIND] = 'E' or ([API-ENTS-PARM.KIND] = 'M' and [API-ENTS-PARM.TYPE1] = 'N') then an interval for entity latest modification times may be specified in API-ENTS-PARM.TIME-FM and API-ENTS-PARM.TIME-TO. Values achieved by resetting do not impose bounds on the times of the entities being selected.

If [API-ENTS-PARM.KIND] = 'E' or ([API-ENTS-PARM.KIND] = 'M' and [API-ENTS-PARM.TYPE1] = 'N'), then a range of user ids may be specified in API-ENTS-PARM.USER. Such specification will restrict the selection of entities to those last modified by users whose ids fit into the range.

If [API-ENTS-PARM.KIND] = 'A' or 'C', then a single status may be specified in API-ENTS-PARM.STATUS.

If [API-ENTS-PARM.KIND] = 'C', then a range of events names may be specified in API-ENTS-PARM.EVENT. This will restrict the selection of PAC objects to those immigrated ([API-ENTS-PARM.STATUS] = ' ') or emigrated ([API-ENTS-PARM.STATUS] NE ' ') with an appropriately named event.

If [API-ENTS-PARM.KIND] = 'A' or 'C', then a single time may be specified in API-ENTS-PARM.CURRENT-ON. The specification will restrict the selection of PAA or PAC objects to those known to be or have been the current (PAA) or the highest numbered non-superseded (PAC) version at the time specified.

If [API-ENTS-PARM.KIND] = 'E' and [API-ENTS-PARM.TYPE1] = 'M', then ranges of origin and destination status names may be specified in [API-ENTS-PARM.STATUS-FM] and [API-ENTS-PARM.STATUS-TO] respectively.

If [API-ENTS-PARM.KIND] = 'C' and [API-ENTS-PARM.TYPE3] = 'B' or 'U', then a single PAC compartment may be specified in API-ENTS-PARM.UAPPLICATION. The specification will restrict the selection of PAC objects to those (known to be) used by (B) or using (U) an object in the specified compartment (of the specified application).

If [API-ENTS-PARM.KIND] = 'A', then an interval of PAA job numbers may be specified in API-ENTS-PARM.JOB-FM and API-ENTS-PARM.JOB-TO The specification will restrict the selection of PAA objects to those belonging to PAA jobs of appropriate numbers. [API-ENTS-PARM.JOB-TO] = 0 does not restrict the selection.

If [API-ENTS-PARM.KIND] = 'A' or ([API-ENTS-PARM.KIND] = 'E' and [API-ENTS-PARM.TYPE1] = 'M'), then a single PAA object state or PAC event state respectively may be specified in API-ENTS-PARM.STATE. The specification will restrict the selection of PAA objects or PAC events to those in the specified state.

PAA object state codes are:

B backed up
C current
H historical
R removed
S scheduled

PAC event state codes are:

A authorized
B backed out
C completed
P pending
S started
U submitted
V validated

[API-ENTS-PARM.LINE-CNT] is made 0, if negative; it is made 60, if greater than 60. It is then interpreted as the number of lines (the size of the chunk) to be returned. In particular, no lines will be returned, if LINE-CNT EQ 0. APINENTS returns in API-ENTS-PARM.LINE-CNT the number of lines of the returned chunk. It may be less than the adjusted original value, if the end of the list has been reached.

Normally, API-ENTS-PARM.LINE-TOT should be reset when the first chunk of a list is wanted and left untouched until the processing of the list is finished. It will then after each call carry the total of list entries already returned.

API-ENTS-PARM.LINE will carry the returned chunk.

Normally, API-ENTS-PARM.API-PAC-AREA should be reset when the first chunk of a list is wanted. API-ENTS-PARM.API-PAC-AREA should be left untouched for a next call to obtain an immediately following chunk of the list.

## API Return Codes

APINENTS returns the following [API-PARM.API-MSG-NO]:

| | |
|---|---|
| 0000 | if a non-final chunk of a list is being returned. |
| 0070 | if no entities satisfying the selection criteria have been found. |
| 1011 | if the specified [API-PARM.API-MAIN-FUNC] is meaningless (ie, is not 'S'). |
| 9999 | if at least one entity satisfying the selection criteria has been found and a final chunk of a list is being returned. |

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.

# 15 Job API

API Routine: APINJOB

Parameters: UGNLPARM, UJBLJOB

The Job API allows users to display (retrieve) information about a particular job. The following function is valid:

D Display job (Retrieve).

This chapter covers the following topics:



## Local Data Area UJBLJOB

The fields to be passed to the API as the second parameter are in the supplied local data area UJBLJOB. The following is a detailed description of these fields:

| Field Name | Format | Description |
|---|---|---|
| JOB | A32 | Name of the job. |
| MOD-TIME | T | Date and time job was last modified. |
| MOD-BY | A8 | User who last modified job. |
| MOD-TID | A8 | Terminal that last modified job. |
| ADD-TIME | T | Date and time job was added. |
| ADD-BY | A8 | User who added job. |
| ADD-TID | A8 | Terminal that added job. |
| NOTES | A60 (1:3) | Comments. |
| USER-VARIABLES | A32 (1:10) | User-defined variables from the job. |

## Job API Return Codes

7250 Job exists.

7251 Job not found.

# 16 APINLCKS

This API lets obtain chunks of lists of locked PAC entities.

APINLCKS can be invoked with:

```
CALLNAT 'APINLCKS' API-PARM API-LCKS-PARM
```

To obtain a chunk, set API-PARM.API-MAIN-FUNC to 'D'. The value of API-PARM.API-SUB-FUNC is ignored. A definition of API-LCKS-PARM is provided in LDA UPILLCKS.

This chapter covers the following topics:

- **Local Data Area UPILLCKS**
- **API Return Codes**

## Local Data Area UPILLCKS

A definition of API-LCKS-PARM is provided in LDA UPILLCKS.

| Field Name | Format | I/O |
|---|---|---|
| API-LCKS-PARM | | |
| LOCK-USER | A9 | in |
| LOCK-TIME-FM | T | in |
| LOCK-TIME-TO | T | in |
| LOCK-FLAGS | A32 | in |
| LOCK-FLAGS | | |
| LOCK-FLAG | L (0:31) | |
| LOCK-APPLICATION | A32 | in |
| LINE-CNT | N3 | in/out |
| LINE-TOT | N10 | out |
| LINE | A250 (1:10) [1*] | in/out |
| LINE | | out |
| LINEA | (1:10) | |
| LCK-USER | A8 | |
| LCK-TID | A8 | |
| LCK-TIME | T | |
| LCK-CLAUSE-1 | A50 | |
| LCK-CLAUSE-2 | A50 | |
| MOD-USER | A8 | |
| MOD-TID | A8 | |

| Field Name | Format | I/O | |
|---|---|---|---|
| MOD-TIME | T | | |
| ENT-CODE-1 | A1 | | |
| ENT-CODE-2 | A5 | | |
| ENT-ID-1 | A32 | | |
| ENT-ID-2 | A32 | | |
| ENT-ID-3 | B8 | | |
| ENT-ID-4 | A5 | | |
| ENT-ID-5 | I2 | | |
| ENT-ID-6 | B4 | | |
| ENT-ID-7 | B4 | | |
| API-PAC-AREA | A64 | in/out | |

[1*]: This is the proposed default value. Any value equal or greater 1 can be defined by the user.

If APINLCKS is called with all "in" parameter fields reset, it returns, chunk by chunk, a list of all locked entities in the current ACF.

The LOCK- fields can be used to restrict the listing of locked entities.

API-LOCK-PARM.LOCK-USER should carry a range of user IDs or be blank. Only entities locked for a user ID within the specified range will be selected.

A blank API-LOCK-PARM.LOCK-USER has the same effect on the selection of entities as it would have if it carried an initial asterisk.

API-LOCK-PARM.TIME-FM should, if not reset, carry a lower bound for the time of the locking of any entity to be selected.

If API-LOCK-PARM.TIME-FM = 0 then the selection of entities is not restricted on the account of this parameter.

API-LOCK-PARM.TIME-TO should, if not reset, carry an upper bound for the time of the locking of any entity to be selected.

If API-LOCK-PARM.TIME-TO = 0 then the selection of entities is not restricted on the account of this parameter.

API-LOCK-PARM.FLAGS is, unless blank, interpreted as an array of switches indicating the kinds of the locked entities to be selected. The correspondence between the numbers of the switches (the [I]s of API-LOCK-PARM.FLAG(I)) and the kinds of entities is the following:

| | |
|---|---|
| 0 | any kind not listed in the remainder of this table |
| 1 | job |
| 2 | object |
| 3 | status |
| 4 | application |
| 5 | event |
| 6 | FTT |
| 7 | application-status link |
| 8 | object-status link |
| 9 | migration path |

API-LOCK-PARM.APPLICATION should, unless blank, carry the name of an application. Only locked entities pertaining to the specified application will be selected. If API-LOCK-PARM.APPLICATION = ' ' then the selection of entities is not restricted on the account of this parameter.

[API-LCKS-PARM.LINE-CNT] is made 0, if negative; it is made 10, if greater than 10. It is then interpreted as the number of lines (the size of the chunk) to be returned. In particular, no lines will be returned, if LINE-CNT = 0. APINLCKS returns in API-LCKS-PARM.LINE-CNT the number of lines of the returned chunk. It may be less than the adjusted original value, if the end of the list has been reached.

When APINLCKS returns the last chunk of the list it also returns, in API-LCKS-PARM.LINE-TOT, the total number of the lines of the list.

API-LCKS-PARM.LINE will carry the returned chunk.

The various subfields of a API-LCKS-PARM.LINE(I) for any I in {1,...,[API-LCKS-PARM.CNT]} are used as follows:

| | |
|---|---|
| LCK-USER | carries the user ID for which the entity is locked. |
| LCK-TID | carries the terminal ID with which the lock is marked. |
| LCK-TIME | carries the time of the locking of the entity. |
| LCK-CLAUSE-1 and LCK-CLAUSE-2 | carry the contents of two fields used by PAC and to link locked entities one to another and indicate the kind of the lock. |
| MOD-USER | carries the user ID with which the entity was last modified. |
| MOD-TID | carries the terminal ID with which the entity was last modified. |
| MOD-TIME | carries the time of the last modification of the entity. |
| ENT-CODE-1 | carries a code of the kind of the entity. The codes used in this field are the following:<br><br>J job<br>V object |

| | |
|---|---|
| | S status<br>A application<br>E event<br>F FTT<br>L application-status link<br>I object-status link<br>T migration path<br>space none of the above |
| ENT-CODE-2 | carries an internal code of the kind of the entity. |
| ENT-ID-1 | carries the name or the first part of the compound name of the entity. The first part of a compound name is application name for an application-status link; it is object name for an object-status link. |
| ENT-ID-2 | carries status name for an application-status link or an object status link. |
| ENT-ID-3 to 7 | may carry some further entity attributes not readily usable by an end-user. |

Normally, API-LCKS-PARM.API-PAC-AREA should be reset when the first chunk of a list of locked entities is wanted. API-LCKS-PARM.API-PAC-AREA should be left untouched for a next call to obtain an immediately following chunk of the same list.

## API Return Codes

APINLCKS returns the following [API-PARM.API-MSG-NO]:

0000 if a non-final chunk of the list of locked entities is being returned.

1011 if the specified [API-PARM.API-MAIN-FUNC] is meaningless (ie, is not 'D').

7205 if the "initialisation" has failed, ie, logical files 210 and 211 have not been found set at a pair of matching ACF and PCF.

7251 if [LOCK-APPLICATION] has been specified but not found to be an application name.

9999 if the final chunk of the list of locked entities is being returned.

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.

# 17 APINLNK

This API lets the user not only obtain data from an application - status link but also create a new application - status link and modify an existing one.

APINLNK replaces APINLINK and can be invoked with:

```
CALLNAT 'APINLNK' API-PARM API-LNK-PARM
```

A To add an application status link

D To obtain data from an application status link

M To modify an application status link

This chapter covers the following topics:

- **Local Data Area USLLLNK**
- **API Return Codes**

## Local Data Area USLLLNK

A definition of API-LNK-PARM is provided in LDA USLLLNK.

| Field Name | Format | I/O |
|---|---|---|
| API-LNK-PARM | | |
| APPLICATION | A32 | in |
| STATUS | A32 | in |
| FILE-TRANS-TABLE | A32 | in/out |
| STEPLIB-CNT | N3 | in/out |
| PRD-CASE-LIB | A32 (1:8) | in/out |
| NEIGHBOUR | A32 | |
| LIBRARY | A32 | out |
| FUSER-DBID | N8 | in/out |
| FUSER-FNR | N5 | in/out |
| FDIC-DBID | N5 | in/out |
| FDIC-FNR | N5 | in/out |
| FPAA-DBID | N5 | in/out |
| FPAA-FNR | N5 | in/out |
| XREF | A1 | in/out |
| ESY-NODE | N5 | in/out |

| Field Name | Format | I/O |
|---|---|---|
| OP-SYSTEM | A8 | out |
| FOREIGN-PDS-CNT | N3 | in/out |
| FOREIGN-PDS | A79 (1:50) | in/out |
| ADD-DATE | T | out |
| ADD-BY | A8 | out |
| ADD-TID | A8 | out |
| MOD-DATE | T | out |
| MOD-BY | A8 | out |
| MOD-TID | A8 | out |
| LNK-USER | A64 | |
| LNK-PASSWORD | A64 | |
| LNK-CODEPAGE | A64 | |

Whenever APINLNK is called:

- API-LNK-PARM.APPLICATION should carry the name of the link's application.
- API-LNK-PARM.STATUS should carry the name of the link's status.

If [API-PARM.API-MAIN-FUNC] = 'A' and the application status link is not found, APINLNK tries to add an application status link with the characteristics specified in the "in/out" fields.

If [API-PARM.API-MAIN-FUNC] = 'D' and the application status link is found, APINLNK returns in the remaining parameter fields the application status link data, ignoring any values put by the user in the "in/out" fields.

If [API-PARM.API-MAIN-FUNC] = 'M' and the application status link is found, APINLNK tries to modify the application status link endowing it with the characteristics specified in the "in/out" fields.

Which of the parameter fields can carry meaningful values for a link depends on the application (Predict or not) and on the type of the status. APINLNK tries to reset all irrelevant parameter fields when called with API-PARM.API-MAIN-FUNC at 'A' or 'M'.

If [API-PARM.API-MAIN-FUNC] is 'M' then [API-LNK-PARM.STEPLIB-CNT] is made 0, if negative.

The names of the step applications are returned or expected to be specified [API-LNK-PARM.STEPLIB-CNT] is the number of step applications (to be) specified in the link.

If [API-PARM.API-MAIN-FUNC] is 'A' or 'M' then [API-LNK-PARM.STEPLIB-CNT] is made 8, if greater than 8. If [API-PARM.API-MAIN-FUNC] is 'A' then a negative [API-LNK-PARM.STEPLIB-CNT] is interpreted as an instruction to make the link inherit the sequence of step applications

specified in the application record. (if API-LNK-PARM.STEPLIB-CNT GT 0) in API-LNK-PARM.STEPLIB(1:[API-LNK-PARM.STEPLIB-CNT]).

A Predict Case library specification in an application status link can be retrieved but not introduced with APINLNK.

If [API-PARM.API-MAIN-FUNC] is 'A' and a relevant [API-LNK-PARM.LIBRARY] is not specified then APINLNK seeks the value among the defaults in the application record. If [API-PARM.API-MAIN-FUNC] is 'A' and a relevant ([...-DBID],[...-FNR]) = (0,0) then APINLNK seeks the values in the status record. If [API-PARM.API-MAIN-FUNC] is 'A' and a relevant [API-LNK-PARM.XREF] is not specified then APINLNK seeks the value in the status record.

The type of operating system is unique for a node and is discovered by APINLNK regardless of any value possibly put in API-LNK-PARM.OP-SYSTEM by the user. APINLNK processes the user-specified values of API-LNK-PARM.FOREIGN-PDS-CNT and API-LNK-PARM.FOREIGN-PDS only if API-LNK-PARM.ESY-NODE not equal to 0. [API-LNK-PARM.FOREIGN-PDS-CNT] is the number of foreign PDSes (to be) specified in the link.

If [API-PARM.API-MAIN-FUNC] is A or M then [API-LNK-PARM.FOREIGN-PDS-CNT] is made 0, if negative, it is made 50, if greater than 50. The foreign PDS specifications are returned or expected to be supplied (if API-LNK-PARM.ESY-NODE not equal to 0 AND API-LNK-PARM.FOREIGN-PDS-CNT greater than 0) in API-LNK-PARM.FOREIGN-PDS(1:[API-LNK-PARM.FOREIGN-PDS-CNT]).

Each element of API-LNK-PARM.FOREIGN-PDS contains or should contain:

- a foreign location type code (bytes 1 - 4), a PDS name (bytes 5 - 58), and operation system specific data:
- (optional) z/OS volume specification (bytes 59 - 64) or (required) BS2000 LMS element type specification (byte 59).

In byte 67 of an element of API-LNK-PARM.FOREIGN-PDS APINLNK puts a S(ource) or L(oadable) marking.

## API Return Codes

APINLNK returns the following [API-PARM.API-MSG-NO]:

1011 If an [API-PARM.API-MAIN-FUNC] distinct from A, D and M has been specified.

6303 If the LMS element type specification is invalid.

7102 If an out-of-range database number is specified.

7103 If and out-of-range file number is specified.

7136 If an attempt has been made to link application Predict to status ARCHIVE or RETIRE.

7250 If the link has been found.

7251 If an application or the status or a foreign type is unknown or the link (to be retrieved) has not been found.

7253 If the link has been added.

7254 If the link has been modified.

7259 If the link (to be added) has not been added for an unspecified reason.

7264 If the link (to be modified) is held.

7267 If more than one location of a single foreign type has been specified.

7428 If an inappropriate [XREF] has been specified.

7517 If Predict or Predict Case has been specified as a step application.

7577 If an attempt has been made to link to a status application Predict Case.

Other values of API-PARM.API-MSG-NO may be received in more unusual situations. APINLNK may pass an Entire System Server message resulting from the ESY's inactivity or an unsuccessful search for a node or a PDS.

Failure to assign to the just added or modified application status link an FTT does not get reflected in API-PARM.API-MSG-NO; it causes the resetting of API-LNK-PARM.FILE-TRANS-TABLE.

# 18 APINLNKS

This API lets obtain a chunk of a list of PAC application - status links.

APINLNKS can be invoked with:

```
CALLNAT 'APINLNKS' API-PARM API-LNKS-PARM
```

To obtain a chunk set API-PARM.API-MAIN-FUNC at 'S'. API-PARM.API-SUB-FUNC is not used by APINLNKS.

This chapter covers the following topics:

- **Local Data Area UPILLNKS**
- **API Return Codes**

## Local Data Area UPILLNKS

A definition of API-LNKS-PARM is provided in LDA UPILLNKS.

| Field Name | Format | Description |
|---|---|---|
| API-LNKS-PARM | | |
| ORDER | A1 | in |
| APPLICATION | A33 | in |
| APPLICATION-TYPE | A16 | in |
| STATUS | A33 | in |
| STATUS-TYPE | A16 | in |
| FILE-TRANS-TABLE | A33 | in |
| STEP-APPL-CNT-FM | N3 | in |
| STEP-APPL-CNT-TO | N3 | in |
| STEP-APPL | A33 (1:8) | in |
| PRD-CASE-LIB | L | in |
| NEIGHBOUR-APPL | A33 | in |
| LIBRARY | A9 | in |
| FUSER-DBNR-COND-K | A1 | in |
| FUSER-DBNR-COND-M | A5 | in |
| FUSER-DBNR-COND-L | N5 | in |
| FUSER-DBNR-COND-U | N5 | in |
| FUSER-FNR-COND-K | A1 | in |
| FUSER-FNR-COND-M | A5 | in |
| FUSER-FNR-COND-L | N5 | in |

| Field Name | Format | Description |
|---|---|---|
| FUSER-FNR-COND-U | N5 | in |
| FDIC-DBNR-COND-K | A1 | in |
| FDIC-DBNR-COND-M | A5 | in |
| FDIC-DBNR-COND-L | N5 | in |
| FDIC-DBNR-COND-U | N5 | in |
| FDIC-FNR-COND-K | A1 | in |
| FDIC-FNR-COND-M | A5 | in |
| FDIC-FNR-COND-L | N5 | in |
| FDIC-FNR-COND-U | N5 | in |
| FPAA-DBNR-COND-K | A1 | in |
| FPAA-DBNR-COND-M | A5 | in |
| FPAA-DBNR-COND-L | N5 | in |
| FPAA-DBNR-COND-U | N5 | in |
| FPAA-FNR-COND-K | A1 | in |
| FPAA-FNR-COND-M | A5 | in |
| FPAA-FNR-COND-L | N5 | in |
| FPAA-FNR-COND-U | N5 | in |
| XREF | A1 | in |
| ESY-NODE-K | A1 | in |
| ESY-NODE-M | A3 | in |
| ESY-NODE-L | N5 | in |
| ESY-NODE-U | N5 | in |
| OP-SYSTEM | A9 | in |
| FOREIGN-PDS-CNT-FM | N3 | in |
| FOREIGN-PDS-CNT-TO | N3 | in |
| FOREIGN-TYPE2 | A4 (1:8) | in |
| FOREIGN-PDS | A55 (1:8) | in |
| FOREIGN-VOL | A7 (1:8) | in |
| FOREIGN-TYPE3 | A1 (1:8) | in |
| ADD-TIME-FM | T | in |
| ADD-TIME-TO | T | in |
| ADD-BY | A9 | in |
| ADD-TID | A9 | in |
| MOD-TIME-FM | T | in |
| MOD-TIME-TO | T | in |
| MOD-BY | A9 | in |

| Field Name | Format | Description |
|---|---|---|
| MOD-TID | A9 | in |
| LINE-CNT | N3 | out |
| LINE-TOT | N10 | in/out |
| LINE | A250 (1:60) | out |
| LINE-STRC | (1:60) | |
| LINE-APPL | A32 | |
| LINE-STAT | A32 | |
| LINE-REST | A186 | |
| API-PAC-AREA | A128 | in/out |

Whenever APINLNKS is called [API-LNKS-PARM.ORDER] must be

A if the links are to be ordered lexicographically by keys wherein application names precede status names,

S if the links are to be ordered lexicographically by keys wherein status names precede application names.

API-LNKS-PARM.APPLICATION may carry a specification of a range that the application names of the links to be selected must fit into.

API-LNKS-PARM.APPLICATION-TYPE may carry a sequence of one character PAC application type codes. The codes are:

D for type PREDICT,

N for type NATURAL,

P for application PREDICT-CASE.

The field should be left blank if links of applications of various types are to participate in the selection.

API-LNKS-PARM.STATUS may carry a spacification of a range that the status names of the links to be selected must fit into.

API-LNKS-PARM.STATUS-TYPE may carry a sequence of one-character PAC status type codes. The codes are:

A for ARCHIVE,

D for development,

I for incorporation,

M for maintenance,

N for neighbour,

P for production,

R for retirement,

T for test.

API-LNKS-PARM.FILE-TRANS-TABLE may carry a range of names specifying the FTTs whose assignment to a link make it eligible for the selection.

An interval for the number of step applications assigned to the eligible links may be specified in API-LNKS-PARM.STEP-APPL-CNT-FM and API-LNKS-PARM.STEP-APPL-CNT-TO. [API-LNKS-PARM.STEP-APPL-CNT-TO] = 0 does not impose a bound on the numbers of step applications.

API-LNKS-PARM.STEP-APPL may carry up to eight ranges of names specifying the PAC applications whose assignment to a link as a step application make the link eligible for the selection. A link is considered eligible on this count if the name of at least one of the step applications it has been assigned fits at least one of the ranges specified. If API-LNKS-PARM.STEP-APPL is completely blank, then no restrictions are imposed on the step application assignments to the links being selected. If API-LNKS-PARM.STEP-APPL has blank elements but is not completely blank, then the blank elements do not affect the eligibility of the links being selected.

If [API-LNKS-PARM.PRD-CASE-LIB] = TRUE, then only links with non-blank PREDICT CASE Library field are returned.

API-LNKS-PARM.NEIGHBOUR-APPL may carry a range of names specifying the PAC applications whose assignment to a link as a neighbour application make the link eligible for the selection. Only links whose statuses are neighbour statuses will occur in a list compiled to specifications with non-blank API-LNKS-PARM.NEIGHBOUR-APPL. If API-LNKS-PARM.NEIGHBOUR-APPL is blank, then no restrictions are imposed on account of this parameter on the links being selected.

API-LNKS-PARM.LIBRARY may carry a range of library names whose occurrence in a link make it eligible for for the selection.

The twenty-four '...-COND-...' fields may carry the specifications of conditions imposed on the data base and file numbers of some of the three files possibly referred to in the links being selected.

A condition is imposed on such a number if the corresponding ...COND-K field carries 'B' (between), 'G' (greater or equal), 'L' (less or equal), 'M' (mask only), 'O' (strictly outside). In any of these cases the number must fit the mask specified in the corresponding ...COND-M field, decimal digits and '.', which will match any digit in the number, being the characters acceptable in a mask. If [...COND-K] = 'M', then the mask is the only condition imposed on the number. In the other cases the number must additionally fit into or, with 'O', outside the specified interval. The lower bound should be specified in ...COND-L and the upper bound in ...COND-U. Thus [...COND-K] = 'L' makes [...COND-U] relevant, and [...COND-K] = 'G' makes [...COND-L] relevant.

API-LNKS-PARM.XREF may carry a character which has to occur in the link's Maintain Xref field for the link to be eligible for the selection. The (possibly) meaningful values of the field are:

L for local,

N for none,

R for remote.

The four ESY-NODE-... fields may carry the specification of conditions imposed on the foreign node numbers referred to in the links being selected. The method of condition specification is the same as with the ...-COND-... fields; only the node numbers have at most three, not five, digits.

API-LNKS-PARM.OP-SYSTEM may carry a range of operating system names which the foreign node referred to in a link must fit for the link to be eligible for the selection.

An interval for the number of foreign locations assigned to the eligible links may be specified in API-LNKS-PARM.FOREIGN-PDS-CNT-FM and API-LNKS-PARM.FOREIGN-PDS-CNT-TO. [API-LNKS-PARM.FOREIGN-PDS-CNT-TO] = 0 does not impose a bound on the numbers of foreign locations.

The (API-LNKS-PARM.FOREIGN-TYPE2,API-LNKS-PARM.FOREIGN-PDS, API-LNKS-PARM.FOREIGN-VOL,API-LNKS-PARM.FOREIGN-TYPE3) quartets may carry up to eight sets of conditions specifying the foreign locations whose assignment to a link make the link eligible for the selection. A link is considered eligible on this count if for each non-blank condition quartet there is at least one foreign location assigned to the link and fitting the quartet. FOREIGN-TYPE2 and FOREIGN-TYPE3 can take single values or remain blank and inactive. FOREIGN-PDS and FOREIGN-VOL can take range specifications.

An interval for the creation times of the links to be selected may be specified in API-LNKS-PARM.ADD-TIME-FM and API-LNKS-PARM.ADD-TIME-TO. Values achieved by resetting do not impose bounds on the times of the links being selected.

API-LNKS-PARM.ADD-BY may carry a range of names specifying the ids of the users who must have added the links to be selected.

API-LNKS-PARM.ADD-TID may carry a range of names specifying the ids of the terminals from which the links to be selected must have been added.

An interval for the latest modification times of the links to be selected may be specified in API-LNKS-PARM.MOD-TIME-FM and API-LNKS-PARM.MOD-TIME-TO. Values achieved by resetting do not impose bounds on the times of the links being selected.

API-LNKS-PARM.MOD-BY may carry a range of names specifying the ids of the users who must have last modified the links to be selected.

API-LNKS-PARM.MOD-TID may carry a range of names specifying the ids of the terminals from which the links to be selected must have been last modified.

[API-LNKS-PARM.LINE-CNT] is made 0, if negative; it is made 60, if greater than 60. It is then interpreted as the number of lines (the size of the chunk) to be returned. In particular, no lines will be returned, if LINE-CNT EQ 0. APINLNKS returns in API-LNKS-PARM.LINE-CNT the number of lines of the returned chunk. It may be less than the adjusted original value, if the end of the list has been reached.

Normally, API-LNKS-PARM.LINE-TOT should be reset when the first chunk of a list is wanted and left untouched until the processing of the list is finished. It will then after each call carry the total of list entries already returned.

API-LNKS-PARM.LINE will carry the returned chunk.

Normally, API-LNKS-PARM.API-PAC-AREA should be reset when the first chunk of a list is wanted. API-LNKS-PARM.API-PAC-AREA should be left untouched for a next call to obtain an immediately following chunk of the list.

## API Return Codes

APINLNKS returns the following [API-PARM.API-MSG-NO]:

| | |
|---|---|
| 0000 | if a non-final chunk of a list is being returned. |
| 0070 | if no entities satisfying the selection criteria have been found. |
| 1011 | if the specified [API-PARM.API-MAIN-FUNC] is meaningless. |
| 7407 | if the specified [API-LNKS-PARM.ORDER] is meaningless. |
| 9999 | if at least one entity satisfying the selection criteria has been found and a final chunk of a list is being returned. |

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.

# 19 Load Object Version Natural Source API

APIs APINLSRC API Routine: APINLSRC

Parameters: UGNLPARM, UOVLLSRC

The Load Object Version Natural Source API allows users to load the Natural source code for a particular object version to Natural source area. The following function is valid:

L Load Natural source to source area.

This chapter covers the following topics:

- **Local Data Area UOVLLSRC**
- **Load Object Version Natural Source API Return Codes**

## Local Data Area UOVLLSRC

The fields to be passed to the API as the second parameter are in the supplied local data area UOVLLSRC. The following is a detailed description of these fields:

| Field Name | Format | Description |
|---|---|---|
| APPLICATION | A32 | Name of the application. |
| OBJECT | A8 | Name of the object version. |
| VERSION-NUM | N4 | Version number of the object version. |

## Load Object Version Natural Source API Return Codes

7251 Object Version not found.

# 20 Locked Objects API

API Routine: APINMEOL

Parameters: UGNLPARM, UMELOLOK

The Locked Objects API allows users to review locked objects during migration event processing. Users can interrogate the object list for a migration event currently being processed after the objects have been locked and before the migration event is unlocked. The following functions are valid:

T Display locked objects for an active migration event.

blank Print the locked objects report.

This chapter covers the following topics:

- **Local Data Area UMELOLOK**
- **Locked Objects API Return Codes**

## Local Data Area UMELOLOK

The fields to be passed to the API as the second parameter are in the supplied local data area UMELOLOK. The following is a detailed description of these fields:

| Field Name | Format | Description | |
|---|---|---|---|
| FUNC | A1 | Requested function. Valid options are | |
| | | T | Display locked objects for the migration event currently being processed. |
| | | blank | Print the locked objects report to print file 1. |
| EVENT-NAME | A32 | Name of the event to be interrogated. | |
| RESP-CODE | P5 | Response code to be returned. | |
| USER-AREA | A50 | Additional area for user parameters. | |

## Locked Objects API Return Codes

1011 Invalid function specified. Invalid event name specified.

7251 Migration event not found.

9999 End of locked objects report processing.

# 21 Migration Event API

API Routine: APINMGEV

Parameters: UGNLPARM, UMELMGEV

The Migration Event API allows users to add, modify, submit, purge or display (retrieve) information about a particular migration event. The following functions are valid:

A Add Migration Event.

D Display Migration Event (Retrieve).

M Modify an existing Migration Event.

P Purge Migration Event.

U Submit Migration Event.

This chapter covers the following topics:



## Local Data Area UMELMGEV

The fields to be passed to the API as the second parameter are in the supplied local data area UMELMGEV. The following is a detailed description of these fields:

| Field Name | Format | Description |
|---|---|---|
| EVENT | A32 | Name of the event. |
| APPLICATION | A32 | Name of the application to be migrated. |
| FM-STATUS | A32 | Name of the status from which the application is to be migrated. |
| TO-STATUS | A32 | Name of the status to which the application is to be migrated. |
| ARCH-EVENT | A32 | Name of the archive event. |
| MAINT-REQ | A20 | Name of the maintenance request for which the migration event was created. |
| REPLACE | A1 | Replace objects (Y/N) at development or maintenance destination status. |
| SCHED-FLAG | A1 | Y indicates that the event is schedules (production status only). |
| SCHED-DATE | T | Earliest date and time this event may be submitted. |
| EXEC-START | T | Date and time the event was started. |
| EXEC-END | T | Date and time the event was ended. |
| STATE-TEXT | A10 | State of the event. |

| Field Name | Format | Description | |
|---|---|---|---|
| STEP-NUMBER | N2 | Current step number of the event. | |
| STEP-TEXT | A16 | Current step of the event. | |
| NATURAL-OBJ-CNT | N5 | Number of Natural objects migrated. | |
| PREDICT-OBJ-CNT | N5 | Number of Predict objects migrated. | |
| MESSAGE-OBJ-CNT | N5 | Number of user error messages migrated. | |
| ERROR-CNT | N5 | Number of errors detected during migration. | |
| NOTES | A60 (1:3) | Notes | |
| GENERATE-TYPE | A1 | Generate List Type. Valid options are | |
| | | P | PAC Generate |
| | | A | Predict Application Generate |
| | | M | Maintenance Request Generate |
| | | S | Predict Set Generate |
| | | H | Automatic Archiving |
| | | L | Locks |
| | | U | PAC generate (from origin status) |
| GEN-SET-USER | A8 | Name of the Predict set used with GENERATE-TYPE S. | |
| GEN-SET-NO | N2 | Predict set number. | |
| GEN-SET-LOC | A1 | Location (local or remote) of the Predict set. | |
| MOD-TIME | T | Date and time event was last modified. | |
| MOD-BY | A8 | User that last modified event. | |
| MOD-TID | A8 | Terminal that last modified event. | |
| ADD-TIME | T | Date and time event was added. | |
| ADD-BY | A8 | User that added event. | |
| GENLIST-U-PARAMETERS | | Optional. Only needed when GENERATE-TYPE=U. | |
| GENLIST-U-URL | A253 | Optional. Only needed when GENERATE-TYPE=U. | |
| GENLIST-U-FILE-PATTERN | A32 | Optional. Only needed when GENERATE-TYPE=U. | |
| GENLIST-U-EXT-PATTERN | A3 | Optional. Only needed when GENERATE-TYPE=U. | |

## Migration Event API Return Codes

7250 Migration event exists.

7251 Migration event not found.

7253 Migration event added.

7254 Migration event has been modified.

7256 Migration event has been purged.

1163 Migration event submitted.

# 22 Maintenance Request API

API Routine: APINMREQ

Parameters: UGNLPARM, UMRLMREQ

The Maintenance Request API allows users to add, modify, purge or display (retrieve) information about a particular maintenance request. The following functions are valid:

A Add Maintenance Request.

D Display Maintenance Request (Retrieve).

M Modify an existing Maintenance Request.

P Purge Maintenance Request.

This chapter covers the following topics:



## Local Data Area UMRLMREQ

The fields to be passed to the API as the second parameter are in the supplied local data area UMRLMREQ. The following is a detailed description of these fields:

| Field Name | Format | Description |
|---|---|---|
| REQUEST | A20 | Name of the maintenance request. |
| NEW-REQUEST | A20 | Not currently implemented. |
| PROBLEM-SUBJECT | A60 | Problem subject. |
| PROBLEM-TYPE | A8 | Problem type. |
| PRIORITY | N3 | Priority of the maintenance request. |
| STATUS | A12 | Current status of the maintenance request. |
| ACTION | A12 | Action to be taken. |
| CONTACT-NAME | A32 | Contact name. |
| CONTACT-TEL | A20 | Contact telephone number. |
| CLOSED-TIMX | T | Date and time maintenance request was closed. |
| CLOSED-BY | A8 | User that closed the maintenance request. |
| CLOSED-TID | A8 | Terminal that closed the maintenance request. |
| RELATED-PROB | A60 | Related problem. |
| NOTES-1 | A60 | Notes line 1. |
| NOTES-2 | A60 | Notes line 2. |

| Field Name | Format | Description |
|---|---|---|
| NOTES-3 | A60 | Notes line 3. |
| MOD-TIME | T | Date and time request was last modified. |
| MOD-BY | A8 | User that last modified request. |
| MOD-TID | A8 | Terminal that last modified request. |
| ADD-TIME | T | Date and time request was added. |
| ADD-BY | A8 | User that added request. |
| ADD-TID | A8 | Terminal that added request. |

## Maintenance Request API Return Codes

1018 Maintenance Request is locked.

7251 Maintenance Request not found.

7253 Maintenance Request has been added successfully.

7254 Maintenance Request has been modified.

7260 Maintenance Request ID missing.

7561 Maintenance Request Problem Subject missing.

7562 Maintenance Request Priority missing.

7563 Maintenance Request Status missing.

7564 Maintenance Request Status not found.

7565 Maintenance Request Action missing.

7566 Maintenance Request Action not found.

7580 Closed Maintenance Request may not be modified.

# 23 APINNSRC

This API lets obtain chunks of

- Natural sources,
- short error messages,
- long error messages,
- foreign sources,
- USED (object) lists, and
- USED BY (object) lists resident in the ACF.

APINNSRC can be invoked with:

```
CALLNAT 'APINNSRC' API-PARM API-NSRC-PARM
```

D To obtain a chunk of a Natural or foreign source or an error message.

U To obtain a chunk of the USED list of a PAC object

B To obtain a chunk of the USED BY list of a PAC object

S The value of API-PARM.API-SUB-FUNC is ignored except when API-PARM.API-MAIN-FUNC=D, in which case [API-PARM.API-SUB-FUNC] = S indicates that a short rather than long error message is required.

This chapter covers the following topics:

- **Local Data Area UOVLNSRC**
- **API Return Codes**

## Local Data Area UOVLNSRC

A definition of API-NSRC-PARM is provided in LDA UOVLNSRC.

| Field Name | Format | I/O |
|---|---|---|
| API-NSRC-PARM | | |
| APPLICATION | A32 | in |
| OBJECT | A32 | in |
| VERSION-NUM | N4 | in/out |
| TYPE | A15 | in/out |
| LINE-CNT | P5 | in/out |
| OTHER-APPLICATION | A32 (1:10) [1*] | out |

| Field Name | Format | I/O |
|---|---|---|
| SOURCE | A94 (1:10) [1*] | out |
| PAC-AREA | B23 | in/out |

[1*]: This is the proposed default value. Any value equal or greater 1 can be defined by the user.

When APINNSRC is called:

API-NSRC-PARM.APPLICATION should carry the name of the application in whose compartment the (versioned) object resides.

- API-NSRC-PARM.OBJECT should carry the name of the object.
- [API-NSRC-PARM.VERSION-NUM], if positive, is interpreted as the version number of the object; if negative or zero, it is interpreted as the highest available version number of an object in the given compartment, with the given (if TYPE not equal to ' ') or chosen (see below) type.
- [API-NSRC-PARM.TYPE], if non-blank, is interpreted as the type designation of the object; if blank, it is interpreted as the lexicographically least available type designation of an object in the given compartment, with the given name.

The possibility to specify the type of the required object is new.

The type designations to be used are the following:

| | |
|---|---|
| NA | Parameter data area |
| NC | Copycode |
| NG | Global data area |
| NH | Helproutine |
| NL | Local data area |
| NM | Map |
| NN | Subprogram |
| NP | Program |
| NS | Subroutine |
| NT | Text |
| EN | Error message |
| VIEW | DDM |
| RULE | Rule |
| AFIA | ADABAS file |
| AFIU | User file |

- For foreign objects, their four-character type designations beginning in '3' can be used.

- [API-NSRC-PARM.LINE-CNT] is made 0, if negative; it is made 10, if greater than 10. It is then interpreted as the number of lines (the size of the chunk) to be returned; no lines will be returned, if LINE-CNT = 0.
- Normally, API-NSRC-PARM.PAC-AREA should be reset when the first chunk of the object's source or list is required and left untouched for a next call to obtain an immediately following chunk of the same source or list. If the (three) directory lines are to be part of the first chunk, H'F0F0F1' should be put in bytes 18 - 20 of API-NSRC-PARM.PAC-AREA.

  If an object satisfying the selection criteria is found, a positive [API-NSRC-PARM.VERSION-NUM] and a non-blank [API-NSRC-PARM.TYPE] are returned.

  APINNSRC returns in API-NSRC-PARM.LINE-CNT the number of lines (the size) of the returned chunk. It may be less than the adjusted original value if the end of the source or list has been reached.
- If API-PARM.API-MAIN-FUNC = 'D' then APINNSCR returns blanks in API-NSRC-PARM.OTHER-APPLICATION(1: API-NSRC-PARM.LINE-CNT).
- If API-PARM.API-MAIN-FUNC = 'U', I GE 1, and I LE API-NSRC-PARM.LINE.CNT then APINNSCR returns a blank in API-NSRC-PARM.OTHER-APPLICATION(I) unless API-NSRC-PARM.SOURCE(I) contains a refernce to an object of an application other than API-NSRC-PARM.APPLICATION, in which case the name of the other application is put into API-NSRC-PARM.OTHER-APPLICATION(I).
- If API-PARM.API-MAIN-FUNC = 'B', I GE 1, and I LE API-NSRC-PARM.LINE.CNT then APINNSCR puts into API-NSRC-PARM.OTHER-APPLICATION(I) the name of the application of the object referred to in API-NSRC-PARM.SOURCE(I).

## API Return Codes

APINNSRC returns the following [API-PARM.API-MSG-NO]:

0000 if the object has been found and a non-final chunk is being returned,

1011 if the specified, ([API-PARM.API-MAIN-FUNC],[API-PARM.API-SUB-FUNC]) is meaningless,

7251 if the object has not been found,

7255 if the object has been found and a final chunk is being returned.

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.

# 24 Object Version API

API Routine: APINOBJV

Parameters: UGNLPARM, UOVLOBJV

The Object Version API allows users to display (retrieve) information about a particular Object Version. The following function is valid:

D Display Object Version (Retrieve).

This chapter covers the following topics:

- **Local Data Area UOVLOBJV**
- **Object Version API Return Codes**

## Local Data Area UOVLOBJV

The fields to be passed to the API as the second parameter are in the supplied local data area UOVLOBJV. The following is a detailed description of these fields:

| Field Name | Format | Description |
|---|---|---|
| APPLICATION | A32 | Name of the application. |
| OBJECT | A32 | Name of the object. |
| VERSION-NUM | N4 | Version number of the object. |
| MAINT-VERSION-NUM | N4 | Version number of the maintenance request. |
| TYPE | A15 | Object type. |
| SAVED-TIME | T | Date and time the object was saved. |
| CAT-TIME | T | Date and time the object was cataloged. |
| MIGRATE-EVENT | A32 | Name of the migration event. |
| CREATE-TIME | T | Create date time. |
| INC-IND | A1 | Incorporation indicator. |
| ARC-IND | A1 | Archive indicator. |
| NAT-LIBRARY | A8 | Natural development directory library. |
| NAT-USER | A8 | Natural development directory user. |
| NAT-SAVE-TIME | T | Natural development directory save date and time. |
| NAT-TP-SYS | A8 | Natural development directory TP system. |
| NAT-TID | A8 | Natural development directory TID. |
| NAT-OP-SYS | A8 | Natural development directory operating system. |
| NAT-TRANS | A8 | Natural development library transaction. |

| Field Name | Format | Description |
|---|---|---|
| NAT-VERS | A6 | Natural development library Natural version. |
| NAT-MODE | A10 | Natural development library mode. |
| NAT-ESIZE | N7 | Natural development library SIZE. |
| PRD-MIGRATE-EVENT | A32 | Name of the Predict migration event. |
| PRD-INIT-TIME | T | Predict migration event initialization date and time. |
| PRD-INIT-USER | A8 | Predict migration event initialization user. |
| PRD-OP-SYS | A8 | Predict migration operating system. |
| PRD-TP-SYS | A8 | Predict migration TP system. |
| PRD-DBID | N5 | Predict migration DBID. |
| PRD-FNR | N5 | Predict migration Fnr. |
| PRD-VERSION | A4 | Predict migration version. |
| PRD-TRUNCATE-CREATOR | A1 | Predict Truncate Creator. |
| PRD-VIEW-DBID | N5 | DBID of View. |
| PRD-VIEW-FNR | N5 | FNR of View. |
| PCA-CREATE-TIME | T | Predict Case creation date and time. |
| PCA-CREATE-USER | A8 | Predict Case creation user. |
| PCA-MAINT-TIME | T | Predict Case maintenance time. |
| PCA-MAINT-USER | A8 | Predict Case maintenance user. |
| PCA-STATUS | A14 | Predict Case status. |
| NUM-SHORT | P2 | Number of user error message short text. |
| NUM-LONG | P2 | Number of user error message long text. |
| LANGUAGES | A1 (1:9) | User error message languages. |
| MOD-TIME | T | Date and time the object version was last modified. |
| MOD-BY | A8 | ID of the user who last modified the object version. |
| MOD-TID | A8 | ID number of terminal at which the object version was last modified. |
| ADD-TIME | T | Date and time the object version was added. |
| ADD-BY | A8 | ID of the user who added the object version. |
| ADD-TID | A8 | ID number of terminal at which the object version was added. |
| API-OBJV-URL | A253 | Optional. URL in external versioning system. |

## Object Version API Return Codes

7251 Object Version not found.

7255 The specified entry has been displayed.

# 25 APINOBLL

API Routine: APINOBLL

Parameters: UGNLPARM, UOLLOBLL

APINOBLL allows users to add, display (retrieve) or modify migration lists containing URLs. The following functions are valid:

A Add URLs to migration list.

D Display (Retrieve) URLs in a migration list.

M Modify URLs in a migration list.

This chapter covers the following topics:

- **Local Data Area UOLLOBLL**
- **Object List API Return Codes**

## Local Data Area UOLLOBLL

The fields to be passed to the API as the second parameter are in the supplied local data area UOLLOBLS. The following is a detailed description of these fields:

| Field Name | Format | Description |
|---|---|---|
| EVENT | A32 | Name of the migration event. |
| LINE-CNT | P5 | Input - number of objects to return. Output - number of objects returned. |
| API-OBLS | A253 (1:10) 1* | Object list. |
| API-OBLS | | |
| API-OBLS-ENTRY | (1:10) | Number of object list entries. |
| OBJECT-NAME | A32 | Name of the object. |
| OBJECT-TYPE | A4 | Type of the object. |
| OBJECT-VERSION | A32 | Version or the status of the object. |
| OBJECT-FILLER | A185 | Object filler. |
| PAC-OBLS-AREA | B20 | Reserved PAC work area. |

[1*]: This is the proposed default value. Any value equal or greater 1 can be defined by the user.

## APINOBLL Return Codes

7250 Object List exists.

7251 Object List not found.

7253 Object List added.

7256 Object List has been purged.

7259 Object List was not added.

# 26 Object List API

**Note:** Use **APINOBLL** instead.

API Routine: APINOBLS

Parameters: UGNLPARM, UOLLOBLS

The Object List API allows users to add, purge or display (retrieve) object list objects for a particular migration event. The following functions are valid:

A Add objects to objects list.

D Display (Retrieve) object list for a migration event.

P Purge the entire object list for a migration event.

This chapter covers the following topics:

- **Local Data Area UOLLOBLS**
- **Object List API Return Codes**

## Local Data Area UOLLOBLS

The fields to be passed to the API as the second parameter are in the supplied local data area UOLLOBLS. The following is a detailed description of these fields:

| Field Name | Format | Description |
|---|---|---|
| EVENT | A32 | Name of the migration event. |
| LINE-CNT | P5 | Input - number of objects to return. Output - number of objects returned. |
| API-OBLS | A80 (1:10) 1* | Object list. |
| API-OBLS | | |
| API-OBLS-ENTRY | (1:10) | Number of object list entries. |
| OBJECT-NAME | A32 | Name of the object. |
| OBJECT-TYPE | A4 | Type of the object. |
| OBJECT-VERSION | A32 | Version or the status of the object. |
| OBJECT-FILLER | A12 | Object filler. |
| PAC-OBLS-AREA | B20 | Reserved PAC work area. |

[1*]: This is the proposed default value. Any value equal or greater 1 can be defined by the user.

## Object List API Return Codes

7250 Object List exists.

7251 Object List not found.

7253 Object List added.

7256 Object List has been purged.

7259 Object List was not added.

# 27 APINOSLS

This API lets obtain a chunk of a list of PAC object - status links.

APINOSLS can be invoked with:

```
CALLNAT 'APINOSLS' API-PARM API-OSLS-PARM
```

To obtain a chunk set API-PARM.API-MAIN-FUNC at 'S'. Set API-PARM.API-SUB-FUNC at ' ', if only current links are wanted; set it at 'H', if historical links too should participate in the selection.

This chapter covers the following topics:

- **Local Data Area UPILOSLS**
- **API Return Codes**

## Local Data Area UPILOSLS

A definition of API-OSLS-PARM is provided in LDA UPILOSLS.

| Field Name | Format | Description |
|---|---|---|
| API-OSLS-PARM | | |
| OBJECT | A33 | in |
| TYPE1 | A1 | in |
| TYPE2 | A16 | in |
| VERSION-FM | N5 | in |
| VERSION-TO | N5 | in |
| APPLICATION | A33 | in |
| APPLICATION-TYPE | A16 | in |
| STATUS | A33 | in |
| STATUS-TYPE | A16 | in |
| EVENT | A33 | in/out |
| ADD-TIME-FM | T | in |
| ADD-TIME-TO | T | in |
| ADD-BY | A9 | in |
| ADD-TID | A9 | in |
| SUP-TIME-FM | T | in |
| SUP-TIME-TO | T | in |
| LINE-CNT | N3 | out |
| LINE-TOT | N10 | in/out |
| LINE | A250 (1:60) [1*] | out |

| Field Name | Format | Description |
|---|---|---|
| LINE-STRC | (1:60) | |
| LINE-NR | B2 | |
| LINE-OBJECT | A32 | |
| LINE-TYPE1 | A1 | |
| LINE-TYPE2 | A4 | |
| LINE-VERSION | N5 | |
| LINE-APPLICATION | A32 | |
| LINE-APPLICATION-TYPE | A1 | |
| LINE-STATUS | A32 | |
| LINE-STATUS-TYPE | A1 | |
| LINE-EVENT | A32 | |
| LINE-ADD-TIME | T | |
| LINE-ADD-BY | A8 | |
| LINE-ADD-TID | A8 | |
| LINE-SUP-TIME | T | |
| LINE-PAD | A78 | |
| API-PAC-AREA | A160 | in/out |

[1*]: This is the proposed default value. Any value equal or greater 1 can be defined by the user.

[API-OSLS-PARM.OBJECT] should carry a specification of a range that the object names of the links to be selected must fit into.

[API-OSLS-PARM.TYPE1] should be:

' ' if the selection is not to be restricted to links of objects of one type1,

D if only links of DDMs are wanted,

E if only links of error messages are wanted,

F if only links of foreign objects are wanted,

N if only links of NATURAL objects are wanted.

If [API-OSLS-PARM.TYPE1] = 'F', then API-OSLS-PARM.TYPF2 may carry the PAC foreign type code of the objects whose links are to be selected. It should be the "3-less" code; eg, 'JCL' rather than '3JCL'.

If [API-OSLS-PARM.TYPE1] = 'N', then API-OSLS-PARM.TYPE2 may carry a sequence of one character external codes of NATURAL types (eg, 'N' for subprogram or 'P' for program).

API-OSLS-PARM.VERSION-FM and API-OSLS-PARM.VERSION-TO may carry the specification of an interval for the PAC version numbers of the objects whose links are to be selected. [API-OSLS-PARM.VERSION-TO] = 0 does not impose an upper bound on the PAC version numbers.

API-OSLS-PARM.APPLICATION may carry a range of names specifying the PAC (ACF) compartments whose objects' links are to participate in the selection.

PI-OSLS-PARM.APPLICATION-TYPE may carry a sequence of one character PAC compartment type codes.

The codes are:

D for type PREDICT,

N for type NATURAL,

P or application PREDICT-CASE.

The field should be left blank if links of objects from compartments of various types are to participate in the selection.

API-OSLS-PARM.STATUS may carry a spacification of a range that the status names of the links to be selected must fit into.

API-OSLS-PARM.STATUS-TYPE may carry a sequence of one character PAC status type codes. The codes of types of statuses for which there can be object - status links are:

A for ARCHIVE,

M for maintenance,

P for production,

T for test.

The field should be left blank if links of statuses of various types are to participate in the selection.

API-OSLS-PARM.EVENT may carry a range of names specifying the migration events which must have created the links to be selected.

An interval for the creation times of the links to be selected may be specified in API-OSLS-PARM.ADD-TIME-FM and API-OSLS-PARM.ADD-TIME-TO. Values achieved by resetting do not impose bounds on the times of the links being selected.

API-OSLS-PARM.ADD-BY may carry a range of names specifying the ids of the users who must have had submitted the migration events which have created the links to be selected.

API-OSLS-PARM.ADD-TID may carry a range of names specifying the ids of the terminals from which the migration events which have created the links to be selected must have been submitted.

An interval for the superseding times of the links to be selected may be specified in API-OSLS-PARM.SUP-TIME-FM and API-OSLS-PARM.SUP-TIME-TO. Values achieved by resetting do not impose bounds on the times of the links being selected.

[API-OSLS-PARM.LINE-CNT] is made 0, if negative; it is made 60, if greater than 60. It is then interpreted as the number of lines (the size of the chunk) to be returned. In particular, no lines will be returned, if LINE-CNT EQ 0. APINOSLS returns in API-OSLS-PARM.LINE-CNT the number of lines of the returned chunk. It may be less than the adjusted original value, if the end of the list has been reached.

Normally, API-OSLS-PARM.LINE-TOT should be reset when the first chunk of a list is wanted and left untouched until the processing of the list is finished. It will then after each call carry the total of list entries already returned.

API-OSLS-PARM.LINE will carry the returned chunk.

Normally, API-OSLS-PARM.API-PAC-AREA should be reset when the first chunk of a list is wanted. API-OSLS-PARM.API-PAC-AREA should be left untouched for a next call to obtain an immediately following chunk of the list.

## API Return Codes

APINOSLS returns the following [API-PARM.API-MSG-NO]:

| | |
|---|---|
| 0000 | if a non-final chunk of a list is being returned. |
| 0070 | if no links satisfying the selection criteria have been found. |
| 1011 | if the specified [API-PARM.API-MAIN-FUNC] or [API-PARM.API-SUB-FUNC] is meaningless. |
| 9999 | if at least one link satisfying the selection criteria has been found and a final chunk of a list is being returned. |

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.

# 28 Migration Path API

API Routine: APINPATH

Parameters: UGNLPARM, UMPLPATH

The Migration Path API allows users to add, modify or display (retrieve) information about a particular migration path. The following functions are valid:

A Add migration path.

D Display migration path (retrieve).

M Modify an existing migration path.

This chapter covers the following topics:

- **Local Data Area UMPLPATH**
- **Migration Path API Return Codes**

## Local Data Area UMPLPATH

The fields to be passed to the API as the second parameter are in the supplied local data area UMPLPATH. The following is a detailed description of these fields:

| Field Name | Format | Description | |
|---|---|---|---|
| APPLICATION | A32 | Name of the application linked to the statuses. | |
| FM-STATUS | A32 | Name of the status from which the application is to be migrated. | |
| TO-STATUS | A32 | Name of the status to which the application is to be migrated. | |
| EXPAND-STATUS | A32 | Name of the status to be used during object list expand. | |
| JOB | A32 | Name of the job that is to perform that migration event. | |
| AUTO-EXPAND | A1 | Autoexpansion option. | |
| BATCH-ONLINE | A1 | Run migration event in batch or online. Valid options are | |
| | | B | Migration event is to be run in batch. |
| | | O | Migration event is to be run online. |
| WORKFILE-USAGE | A1 | Work file usage. Valid options are | |
| | | Y | Work file is to be used during the processing of migration events, valid for batch migrations only. |
| | | N | Default. |
| COPY-MOVE-INCLUDE | A1 | Copy, move, or include objects: | |
| | | C | Objects are to be copied, but still exist in the origin location. |

| Field Name | Format | Description | |
|---|---|---|---|
| | | M | Objects are to be moved from the origin location with no copy to exist in the origin location. `Move' is only valid from a development, test, or maintenance status. |
| | | I | Objects are to be included from a work file. `Include' is only valid for batch migrations from a development or incorporation status. |
| SEPARATE-AUTHORITY | A1 | Authorizer of the event who is different from the person who created the event. | |
| AUTHORIZERS | A8 (1:8) | ID of the user or group who can authorize the event. | |
| ADD-TIME | T | Date and time the migration path was added. | |
| ADD-BY | A8 | ID of the user who added the migration path. | |
| ADD-TID | A8 | ID of the terminal at which the migration path was added. | |
| MOD-TIME | T | Date and time the migration path was last modified. | |
| MOD-BY | A8 | ID or the user who last modified the migration path. | |
| MOD-TID | A8 | ID of the terminal at which the migration path was last modified. | |

## Migration Path API Return Codes

7250 Migration path exists.

7251 Migration path not found.

7253 Migration path added.

7254 Migration path has been modified.

# 29 APINPRF

This API lets obtain data from a PAC profile, create a new profile, and modify an existing one.

APINPRF can be invoked with:

```
CALLNAT 'APINPRF' API-PARM API-PRF-PARM
```

A definition of API-PARM is provided in LDA UGNLPARM in library SYSPACUS.

A add a profile
D obtain data from a profile
M modify a profile

This chapter covers the following topics:

- **Local Data Area UPILPRF**
- **API Return Codes**

## Local Data Area UPILPRF

A definition of API-PRF-PARM is provided in LDA UPILPRF in library SYSPACUS.

| Field Name | Format | I/O |
|---|---|---|
| API-PRF-PARM | | |
| PROFILE | A8 | in |
| NAME | A32 | in/out |
| BATCH-ID | A8 | in/out |
| LIST-LOCKED | A1 | in/out |
| RIGHTS | A20 | in/out |
| ADMINISTRATION | A1 | |
| AUTHORISATION | A1 | |
| SUBMISSION | A1 | |
| 2X | A1 | |
| APPLICATION | A1 | |
| LINK | A1 | |
| CCL | A1 | |
| FTT | A1 | |
| JCL-TEXT | A1 | |
| KEYWORD | A1 | |

| Field Name | Format | I/O |
|---|---|---|
| MAINT-REQUEST | A1 | |
| EVENT | A1 | |
| PATH | A1 | |
| OBJECT | A1 | |
| STATUS | A1 | |
| ADD-DATE | T | out |
| ADD-BY | A8 | out |
| ADD-TID | A8 | out |
| MOD-DATE | T | out |
| MOD-BY | A8 | out |
| MOD-TID | A8 | out |

If [API-PARM.API-MAIN-FUNC] = 'A' and the profile is not found, APINPRF tries to add a profile with the characteristics specified in the "in/out" fields.

If [API-PARM.API-MAIN-FUNC] = 'D' and the profile is found, APINPRF returns in the remaining parameter fields the profile data, ignoring any values put by the user in the "in/out" fields.

If [API-PARM.API-MAIN-FUNC] = 'M' and the profile is found, APINPRF tries to modify the profile endowing it with the characteristics specified in the "in/out" fields.

The meaningful values of the "in/out" fields are listed in the following table.

| | | | | |
|---|---|---|---|---|
| NAME | A32 | any | | |
| BATCH-ID | A8 | any | | |
| LIST-LOCKED | A1 | Y, N | | |
| ADMINISTRATION | A1 | Y, N | | |
| AUTHORISATION | A1 | Y, N | | |
| SUBMISSION | A1 | Y, N | | |
| APPLICATION | A1 | M, R, N | | |
| LINK | A1 | M, R, N | | |
| CCL | A1 | R, N | | |
| FTT | A1 | M, R, N | | |
| JCL-TEXT | A1 | M, R, N | | |
| KEYWORD | A1 | M, R, N | | |
| MAINT-REQUEST | A1 | M, R, N | | |
| EVENT | A1 | M, R, N | | |
| PATH | A1 | M, R, N | | |
| OBJECT | A1 | R, N | | |

| | | | | |
|---|---|---|---|---|
| STATUS | A1 | M, R, N | | |

The meanings of the values are the same as those of their menu system counterparts.

## API Return Codes

APINPRF returns the following [API-PARM.API-MSG-NO]:

1011 if an [API-PARM.API-MAIN-FUNC] distinct from A, D and M has been specified,

7250 if the profile has been found,

7251 if the profile has not been found,

7253 if the profile has been added,

7254 if the profile has been modified,

7264 if the profile (to be modified) is held,

7407 if a meaningless [API-PRF-PARM.LIST-LOCKED] has been specified,

7600 if a meaningless [API-PRF-PARM.ADMINISTRATION], [API-PRF-PARM.AUTHORISATION], or [API-PRF-PARM.SUBMISSION] has been specified,

7601 if a meaningless value has been specified for any of the redefined members of [API-PRF-PARM.RIGHTS] not covered by 7600.

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.

# 30 APINPRFS

This PAC API lets obtain chunks of lists of user ids for which there are PAC profiles.

APINPRFS corresponds to the Select Profiles function of the menu system. Unlike them, it produces only the user ids, including SYSTEM: the name of the default profile. These user IDs can then be used as input to APINPRF which in turn corresponds to the Add, Display, and Modify Profile functions of the menu system.

APINPRFS can be invoked with:

```
CALLNAT 'APINPRFS' API-PARM API-PRFS-PARM
```

A definition of API-PARM is provided in LDA UGNLPARM in library SYSPACUS.

S obtain a chunk of a list of user IDs.

The value of API-PARM.API-SUB-FUNC is ignored.

This chapter covers the following topics:

- **Local Data Area UPILPRFS**
- **API Return Codes**

## Local Data Area UPILPRFS

A definition of API-PRFS-PARM is provided in LDA UPILPRFS in library SYSPACUS.

| Field Name | Format | I/O |
|---|---|---|
| API-PRFS-PARM | | |
| PROFILE | A9 | in |
| PROF-CNT | N3 | in/out |
| PROF | A8 (1:60) [1*] | in/out |
| API-PAC-AREA | A145 | in/out |

[1*]: This is the proposed default value. Any value equal or greater 1 can be defined by the user.

When APINPRFS is called:

- API-PRFS-PARM.PROFILE should carry a range of profile names or be blank. A blank API-PRFS-PARM.PROFILE has the same effect on the selection as one carrying an initial asterisk.
- [API-PRFS-PARM.PROF-CNT] is made 0, if negative; it is made 60, if greater than 60. It is then interpreted as the maximum number of user ids to be returned; no user ids will be returned, if API-PRFS-PARM.PROF-CNT = 0.

- Normally, API-PRFS-PARM.API-PAC-AREA should be reset when the firstchunk of a list of user ids is wanted. API-PRFS-PARM.API-PAC-AREA should be left untouched for a next call to obtain an immediately following chunk of the same list.
- APINPRFS returns in API-PRFS-PARM.PROF-CNT the number of user ids in the returned chunk; it may be less than the adjusted original value, if the end of the user id list has been reached. If no profile-owning user id satisfying the selection criteria is found or a non-positive [API-PRFS-PARM.PROF-CNT] has been specified then AINPRFS returns 0 in API-PRFS-PARM.PROF-CNT.
- If APINPRFS returns a non-zero [API-PRFS-PARM.PROF-CNT] then it returns the chunk of the list of the profile-owning user ids satisfying the selection criteria in API-PRFS-PARM.PROF(1:[API-PRFS-PARM.PROF-CNT]).

## API Return Codes

APINPRFS returns the following [API-PARM.API-MSG-NO]:

0000 if at least one profile-owning user id satisfying the selection criteria has been found and a non-final chunk of a list of the profile-owning user ids satisfying the selection criteria is being returned,

1011 if [API-PARM.API-MAIN-FUNC] other than 'S' has been specified,

7251 if profile-owning user ids satisfying the selection criteria have been sought but none has been found,

9999 if at least one profile-owning user id satisfying the selection criteria has been found and a final chunk of a list of the profile-owning user ids satisfying the selectioncriteria is being returned.

# 31 Selection API

API Routine: APINSEL

Parameters: UGNLPARM, UGNLSEL

The Selection API allows users to select PAC objects with specific selection criteria. The following functions are available:

A Select an Application.

E Select a Migration Event.

H Select an Archive Event.

J Select a Job.

L Select an Application Status Links.

M Select a Maintenance Request.

S Select a Status.

T Select a Migration Path.

V Select a Versioned Object.

This chapter covers the following topics:

- **Local Data Area UGNLSEL**
- **Selection API Return Codes**

## Local Data Area UGNLSEL

A definition of API-SEL-PARM is provided in LDA UGNLSEL.

APINSEL now returns 7251 rather than 7250 in API-PARM.API-MSG-NO if an existing application has been specified and there are no versioned objects or change control logs satisfying all selection criteria.

| Field Name | Format | I/O |
|---|---|---|
| API-SEL-PARM | | |
| CRIT-1 | A32 | in |
| CRIT-2 | A32 | in |
| CRIT-3 | A32 | in |
| OBJ-CNT | P5 | in/out |
| OBJ-1 | A32 (1:10) [1*] | out |
| OBJ-1 | | |

| Field Name | Format | I/O |
|---|---|---|
| OBJ-1-ENTRY | (1:10) | |
| CCL-PGM | A8 | |
| CCL-VER | N4 | |
| CCL-MREQ | A20 | |
| OBJ-1 | | |
| OBJ-1-ENTRY-V | (1:10) | |
| OBJV-OBJECT-ID | A32 | |
| OBJ-2 | A32 (1:10) [1*] | out |
| OBJ-2 | | |
| OBJ-2-ENTRY | (1:10) | |
| CCL-LIB | A8 | |
| CCL-DB | N3 | |
| CCL-FNR | N3 | |
| CCL-USER | A8 | |
| CCL-STATE | A3 | |
| CCL-DATE | T | |
| OBJ-2 | | |
| OBJ-2-ENTRY-V | (1:10) | |
| OBJV-VERSION | A4 | |
| OBJV-TYPE | A6 | |
| PAC-AREA | B126 | in/out |

[1*]: This is the proposed default value. Any value equal or greater 1 can be defined by the user.

## Selection API Return Codes

1011 Invalid function.

7251 Object not found.

9999 End of selection.

# 32 APINSELE

This API lets obtain chunks of lists of event names.

APINSELE corresponds to the Select Events function of the menu system. It produces only the names of the events. These names can then be used as input to APINAUTH, APINDSC, APINMEOL, APINMGEV, APINOBLS, or APINTEXT.

APINSELE can be invoked with:

```
CALLNAT 'APINSELE' API-PARM API-SELE-PARM
```

S obtain a chunk of a list of event names.

The value of API-PARM.API-SUB-FUNC is ignored.

This chapter covers the following topics:

## Local Data Area UPILSELE

A definition of API-SELE-PARM is provided in LDA UPILSELE.

| Field Name | Format | I/O |
|---|---|---|
| API-SELE-PARM | - | - |
| EVENT | A33 | in |
| APPLICATION | A33 | in |
| STATUS-FM-TYPE | A1 | in |
| STATUS-FM | A33 | in |
| STATUS-TO-TYPE | A1 | in |
| STATUS-TO | A33 | in |
| EXECUTION | A1 | in |
| JCL-TEXT-NAME | A33 | in |
| STATE | A1 | in |
| NATURAL-COUNT-FM | N8 | in |
| NATURAL-COUNT-TO | N8 | in |
| ERRMESS-COUNT-FM | N8 | in |
| ERRMESS-COUNT-TO | N8 | in |
| PREDICT-COUNT-FM | N8 | in |
| PREDICT-COUNT-TO | N8 | in |
| FOREIGN-COUNT-FM | N8 | in |
| FOREIGN-COUNT-TO | N8 | in |

| Field Name | Format | I/O |
|---|---|---|
| USER-MODI | A9 | in |
| USER-AUTH | A9 | in |
| USER-SUBM | A9 | in |
| USER-HOLD | A9 | in |
| TIME-MODI-FM | T | in |
| TIME-MODI-TO | T | in |
| TIME-SCHE-FM | T | in |
| TIME-SCHE-TO | T | in |
| TIME-AUTH-FM | T | in |
| TIME-AUTH-TO | T | in |
| TIME-SUBM-FM | T | in |
| TIME-SUBM-TO | T | in |
| TIME-COMP-FM | T | in |
| TIME-COMP-TO | T | in |
| TIME-BACK-FM | T | in |
| TIME-BACK-TO | T | in |
| EV-CNT | N3 | in/out |
| EV | A32 (1:60) [1*] | out |
| API-PAC-AREA | A145 | in/out |

[1*]: This is the proposed default value. Any value equal or greater 1 can be defined by the user.

When APINSELE is called:

- API-SELE-PARM.EVENT should be blank or carry a range of names of the events to be selected.
- API-SELE-PARM.APPLICATION should be blank or carry a range of names of the applications to which events must belong to be selected.
- API-SELE-PARM.STATUS-FM-TYPE should carry a character designating the type(s) of the origin statuses of the events to be selected. The following characters designate status types:

| | |
|---|---|
| A | type of ARCHIVE |
| C | type of CONTROL |
| D | development |
| I | incorporation |
| M | maintenance |
| P | production |
| T | test |
| * | any |

| | |
|---|---|
| space | any |

- API-SELE-PARM.STATUS-FM should be blank or carry a range of names of the statuses that may be the origin statuses of the events to be selected.
- API-SELE-PARM.STATUS-TO-TYPE should carry a character designating the type(s) of the destination statuses of the events to be selected.
- API-SELE-PARM.STATUS-TO should be blank or carry a range of names of the statuses that may be the destination statuses of the events to be selected.
- API-SELE-PARM.EXECUTION should carry a character designating the mode of execution of the events to be selected. The following characters designate modes of event execution:

| | |
|---|---|
| B | batch |
| O | online |
| * | any |
| space | any |

- API-SELE-PARM.JCL-TEXT-NAME should be blank or carry a range of names of the JCL texts used by the events to be selected.
- API-SELE-PARM.STATE should carry a character designating the state(s) of the events to be selected. The following characters designate job states.

| | |
|---|---|
| A | authorised |
| B | backed out |
| C | completed |
| P | pending |
| S | started |
| U | submitted |
| V | validated |
| * | any |
| space | any |

- API-SELE-PARM.NATURAL-COUNT-FM should carry a lower bound of the numbers of NATURAL objects that have participated in the migrations defined by the events to be selected.
- API-SELE-PARM.NATURAL-COUNT-TO should carry 0 or an upper bound of the numbers of NATURAL objects that have participated in the migrations defined by the events to be selected. If API-SELE-PARM.NATURAL-COUNT-TO = 0 then the selection of events is not restricted on the account of this parameter.
- API-SELE-PARM.ERRMESS-COUNT-FM should carry a lower bound of the numbers (counts) of error messages that have participated in the migrations defined by the events to be selected.

- API-SELE-PARM.ERRMESS-COUNT-TO should carry 0 or an upper bound of the numbers (counts) of error messages that have participated in the migrations defined by the events to be selected.

  If API-SELE-PARM.ERRMESS-COUNT-TO = 0 then the selection of events is not restricted on the account of this parameter.
- API-SELE-PARM.PREDICT-COUNT-FM should carry a lower bound of the numbers of PREDICT objects or DDMs that have participated in the migrations defined by the events to be selected.
- API-SELE-PARM.PREDICT-COUNT-TO should carry 0 or an upper bound of the numbers of PREDICT objects or DDMs that have participated in the migrations defined by the events to be selected.

  If API-SELE-PARM.PREDICT-COUNT-TO = 0 then the selection of events is not restricted on the account of this parameter.
- API-SELE-PARM.FOREIGN-COUNT-FM should carry a lower bound of the numbers of foreign objects that have participated in the migrations defined by the events to be selected.
- API-SELE-PARM.FOREIGN-COUNT-TO should carry 0 or an upper bound of the numbers of foreign objects that have participated in the migrations defined by the events to be selected. If API-SELE-PARM.FOREIGN-COUNT-TO = 0 then the selection of events is not restricted on the account of this parameter.
- API-SELE-PARM.USER-MODI should be blank or carry a range of ids of the users who have last modified the events to be selected.
- API-SELE-PARM.USER-AUTH should be blank or carry a range of ids of the users who have last authorised the events to be selected.
- API-SELE-PARM.USER-SUBM should be blank or carry a range of ids of the users who have last submitted the events to be selected.
- API-SELE-PARM.USER-HOLD should be blank or carry a range of ids of the users by whom the events to be selected remain locked.
- API-SELE-PARM.TIME-MODI-FM should carry a lower bound of the modification times of the events to be selected.
- API-SELE-PARM.TIME-MODI-TO should carry an upper bound of the modification times of the events to be selected or 0.

  If API-SELE-PARM.TIME-MODI-TO = 0 then the selection of events is not restricted on the account of this parameter.
- API-SELE-PARM.TIME-SCHE-FM should carry a lower bound of the schedule times of the events to be selected.
- API-SELE-PARM.TIME-SCHE-TO should carry an upper bound of the schedule times of the events to be selected or 0. If API-SELE-PARM.TIME-SCHE-TO = 0 then the selection of events is not restricted on the account of this parameter.

- API-SELE-PARM.TIME-AUTH-FM should carry a lower bound of the authorisation times of the events to be selected.
- API-SELE-PARM.TIME-AUTH-TO should carry an upper bound of the authorisation times of the events to be selected or 0.

  If API-SELE-PARM.TIME-AUTH-TO = 0 then the selection of events is not restricted on the account of this parameter.
- API-SELE-PARM.TIME-SUBM-FM should carry a lower bound of the submission times of the events to be selected.
- API-SELE-PARM.TIME-SUBM-TO should carry an upper bound of the submission times of the events to be selected or 0.

  If API-SELE-PARM.TIME-SUBM-TO = 0 then the selection of events is not restricted on the account of this parameter.
- API-SELE-PARM.TIME-COMP-FM should carry a lower bound of the completion times of the events to be selected.
- API-SELE-PARM.TIME-COMP-TO should carry an upper bound of the completion times of the events to be selected or 0. If API-SELE-PARM.TIME-COMP-TO = 0 then the selection of events is not restricted on the account of this parameter.
- [API-SELE-PARM.EV-CNT] is made 0, if negative; it is made 60, if greater than 60. It is then interpreted as the maximum number of event names to be returned; no event names will be returned, if API-SELE-PARM.EV-CNT = 0.
- Normally, API-SELE-PARM.PAC-AREA should be reset when the first chunk of a list of event names is wanted. API-SELE-PARM.PAC-AREA should be left untouched for a next call to obtain an immediately following chunk of the same list.

APINSELE returns in API-SELE-PARM.EV-CNT the number of event names in the returned chunk; it may be less than the adjusted original value, if the end of the event name list has been reached.

If no job satisfying the selection criteria is found or a non-positive [API-SELE-PARM.EV-CNT] has been specified then APINSELE returns 0 in API-SELE-PARM.EV-CNT.

If APINSELE returns a non-zero [API-SELE-PARM.EV-CNT] then it returns the chunk of the list of the names of the events satisfying the selection criteria in API-SELE-PARM.NUMBER(1:[API-SELE-PARM.EV-CNT]).

## API Return Codes

APINSELE returns the following [API-PARM.API-MSG-NO]:

0000 if at least one event satisfying the selection criteria has been found and a non-final chunk of a list of the names of the events satisfying the selection criteria is being returned,

1011 if [API-PARM.API-MAIN-FUNC] other than 'S' has been specified,

7251 if events satisfying the selection criteria have been sought but none has been found,

9999 if at least one event satisfying the selection criteria has been found and a final chunk of a list of the names of the events satisfying the selection criteria is being returned.

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.

# 33 APINSELO

This API lets the user obtain chunks of lists of current and historical object status links.

APINSELO can be invoked with:

```
CALLNAT 'APINSELO' API-PARM API-OBJECT-STATUS API-PAC-AREA
```

The API allows both MAIN-FUNCTION parameters as well as SUB-FUNCTION parameters.

MAIN-FUNCTION parameter:

N get a list of objects not assigned to a status

O obtain a chunk of a list of objects belonging to a given application and migrated to a given test, production, or maintenance deployment

S obtain a chunk of a list of object status links for any, or one given, application and any status.

SUB-FUNCTION parameters:

H get a list of both the current and the historical object status links satisfying the selection criteria

blank get a list of only the current object status links satisfying the selection criteria

This chapter covers the following topics:

- **Local Data Area UGNASELO**
- **API Return Codes**

## Local Data Area UGNASELO

A definition of API-OBJECT-STATUS is provided in PDA .

| Field Name | Format | I/O |
|---|---|---|
| API-OBJECT-STATUS | | |
| API-APPLICATION | A32 | in |
| API-STATUS | A32 | in |
| API-SEL-OBJECT | A32 | in |
| API-OBJECT-LIST | A32 (10) [1*] | out |
| API-OBJVER-LIST | N5 (10) [1*] | out |
| API-OBJTYP-LIST | A6 (10) [1*] | out |
| API-APPLICATION-LIST | A32 (10) [1*] | out |
| API-STATUS-LIST | A32 (10) [1*] | out |

| Field Name | Format | I/O |
|---|---|---|
| API-TIME-IN-LIST | T (10) [1*] | out |
| API-TIME-OU-LIST | T (10) [1*] | out |
| API-COUNT | N2 | out |
| API-PAC-AREA | B126 | out |

[1*]: This is the proposed default value. Any value equal or greater 1 can be defined by the user.

When APINSELO is called:

- API-OBJECT-STATUS.API-SEL-OBJECT should carry an object name specification. '*', '<', or '>' may be used.
- A single application must be specified in API-OBJECT-STATUS.API-APPLICATION if [API-PARM.API-MAIN-FUNC] = 'O'.
- A single application may be specified in API-OBJECT-STATUS.API-APPLICATION if [API-PARM.API-MAIN-FUNC] = 'S'.

  Please note that inter-application searches may take very long.
- A single status must be specified in API-OBJECT-STATUS.API-STATUS if [API-PARM.API-MAIN-FUNC] = 'O'; the field's value is ignored if [API-PARM.API-MAIN-FUNC] = 'S'.
- Normally, API-OBJECT-STATUS.API-PAC-AREA should be reset when the first chunk of a list is required and left untouched for a next call to obtain the following chunk of the same list.
- To select objects that have no status, set [API-PARM.API-MAIN-FUNC] = 'N' and [API-PARM.API-SUB-FUNC] = blank. This will produce a list of objects that have no status.

APINSELO returns in API-OBJECT-STATUS.API-COUNT the number of object status links in the returned chunk.

The [API-OBJECT-STATUS.API-TIME-IN-LIST] of a link is the time of the versioned object's emigration to the deployment.

The [API-OBJECT-STATUS.API-TIME-OU-LIST] of a link is the time at which a PAC migration was run to remove the object from, or supersede it in, the deployment. This value is non-null only if the link is historical.

## API Return Codes

APINSELO returns the following [API-PARM.API-MSG-NO]:

0000 if object - status links satisfying the selection criteria have been found and a chunk of their list is being returned,

1011 if the specified ([API-PARM.API-MAIN-FUNC],[API-PARM.API-SUB-FUNC]) is meaningless,

7325 if no object - status links have been found to satisfy the selection criteria,

9999 only if object status links satisfying the selection criteria have been found and the final chunk of their list is being returned.

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.

# 34 APINSEVT

This API allows you to submit a batch event and specify the values of user-prompted substitution parameters.

APINSEVT can be invoked with:

```
CALLNAT 'APINSEVT' API-PARM API-SEVT-PARM
```

A definition of API-PARM is provided in LDA UGNLPARM in library SYSPACUS.

To submit an event set [API-PARM.API-MAIN-FUNC] = 'U'.

This chapter covers the following topics:

- **Local Data Area UPILSEVT**
- **API Return Codes**

## Local Data Area UPILSEVT

A definition of API-SEVT-PARM is provided in LDA UPILSEVT in library SYSPACUS.

| Field Name | Format | Description |
|---|---|---|
| API-SEVT-PARM | | |
| EVENT | A32 | Name of the event. |
| SUBS-PARM | A32 (1:10) | Name(s) of substitution parameter(s). |
| SUBS-VAL | A32 (1:10) | Value(s) assigned to the corresponding substitution parameter(s). |

Whenever APINSEVT is called, API-SEVT-PARM.EVENT should carry the name of the event.

If [API-PARM.API-MAIN-FUNC] = 'U', the event is found and authorised, and the user has the permissions required for submitting the event, then an attempt is made to submit the event, assigning to each user-prompted substitution parameter named in an API-SEVT-PARM.SUBS-PARM(I) the value specified in API-SEVT-PARM.SUBS-VAL(I).

The names of the user-prompted substitution parameters should be specified without the JCL substitution character.

## API Return Codes

APINSEVT returns the following [API-PARM.API-MSG-NO]:

1011 if an [API-PARM.API-MAIN-FUNC] distinct from 'U' has been specified.

1163 if the event has been submitted.

2363 if the user has not the permissions needed for submitting the event.

7011 if the event has not been authorized.

7021 if the locks on PAC entities currently in effect preclude the submission of the event.

7024 if the event has already been submitted and is past the state allowing another submission.

7134 if the JCL text to be used for the event's submission has not been found.

7205 if the "initialization" has failed, i.e., logical files 210 and 211 have not been found set at a pair of matching ACF and PCF.

7249 if the event is an online one.

7251 if the event has not been found.

7264 if the event record is held.

7273 if the event is not one "to production", is scheduled, and the scheduled time has not been reached yet.

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.

# 35 Status API

API Routine: APINSTAT

Parameters: UGNLPARM, UAPLSTAT

The Status API allows users to add, modify or display (retrieve) information about a particular status. The following functions are valid:

A Add status.

D Display status (Retrieve).

M Modify an existing status.

This chapter covers the following topics:

- **Local Data Area UAPLSTAT**
- **Status API Return Codes**

## Local Data Area UAPLSTAT

The fields to be passed to the API as the second parameter are in the supplied local data area UAPLSTAT. The following is a detailed description of these fields:

| Field Name | Format | Description | |
|---|---|---|---|
| STATUS | A32 | Name of the Status. | |
| STATUS-TYPE | A2 | Type of the Status. Valid Status types are: | |
| | | A | Archive |
| | | C | Control |
| | | D | Development |
| | | H | History |
| | | I | Incorporation |
| | | M | Maintenance |
| | | P | Production |
| | | R | Retire |
| | | T | Test |
| FUSER-DBID | N5 | Database ID number where the application is located. | |
| FUSER-FNR | N5 | File number where the application is located. | |
| FDIC-DBID | N5 | Database ID number where the XREF information is located. | |
| FDIC-FNR | N5 | File number where the XREF information is located. | |
| XREF | A1 | Local or remote XREF. | |

| Field Name | Format | Description |
|---|---|---|
| MOD-TIME | T | Date and time status was last modified. |
| MOD-BY | A8 | User that last modified status. |
| MOD-TID | A8 | Terminal that last modified status. |
| ADD-TIME | T | Date and time status was added. |
| ADD-BY | A8 | User that added status. |
| ADD-TID | A8 | Terminal that added status. |
| NOTES | A60 (1:3) | Comments. |

## Status API Return Codes

7250 Status exists.

7251 Status not found.

7253 Status added.

7254 Status has been modified.

# 36 APINSTFF

API Routine: APINSTFF

This API lets obtain from the ACF a chunk of:

- a Natural source,
- a Natural loadable,
- a foreign object (be it a source or a loadable),
- a DDM,
- a short error message (seen as a sequence of up to 60 lines, one per language),
- long error messages (those of same number and different languages stringed one after another),
- a USED list,
- a USED BY list.

**Note:** APINSTFF can be seen as APINNSRC expanded and corrected.

APINSTFF can be invoked with:

```
CALLNAT 'APINSTFF' API-PARM API-STFF-PARM
```

Set API-PARM.API-MAIN-FUNC to:

D To obtain a chunk.
S If the chunk is to be one of a short error message.Otherwise leave it blank.

A definition of API-STFF-PARM is provided in LDA UPILSTFF in library SYSPACUS.

This chapter covers the following topics:

## Local Data Area UPILSTFF

| Field Name | Format | Description |
|---|---|---|
| API-STFF-PARM | | |
| APPLICATION | A32 | in |
| OBJECT | A32 | in |
| VERSION | N4 | in/out |
| TYPE | A6 | in/out |
| TYPE1 | A1 | |

| Field Name | Format | Description |
|---|---|---|
| TYPE2 | A4 | |
| TYPE3 | A1 | |
| LINE-CNT | P5 | in/out |
| OTHER-APPLICATION | A32 (1:10) [1*] | out |
| LINE | A250 (1:10) [1*] | out |
| API-PAC-AREA | A64 | in/out |

[1*]: This is the proposed default value. Any value equal or greater 1 can be defined by the user.

Whenever APINSTFF is called API-STFF-PARM.APPLICATION should carry the name of the compartment of tha ACF holding the object and API-STFF-PARM.OBJECT should carry the name of the object.

API-STFF-PARM.VERSION may carry the version number of the object. If API-STFF-PARM.VERSION LE 0, then the highest numbered of the otherwise suitable is chosen; in this case the number of the chosen version is returned in API-STFF-PARM.VERSION.

[API-STFF-PARM.TYPE1] must be 'D' for DDM, 'E' for error message, 'F' for foreign, or 'N' for NATURAL.

The four possible settings of API-STFF-PARM.TYPE1 can be usefully combined with the following values of API-STFF-PARM.TYPE3:

| TYPE1 | TYPE3 |
|---|---|
| D | ' ' for the DDM as normally stored in an FDIC, |
| | 'B' for the USED BY list. |
| E | ' ' for the short or long error messages themselves, |
| | 'B' for the USED BY list. |
| F | 'S' for source, |
| | 'L' for loadable. |
| N | 'B' for the USED BY list, |
| | 'L' for loadable, 'S' for source, 'U' for USED list. |

If [API-STFF-PARM.TYPE1] = 'F', then API-STFF-PARM.TYPE2 should carry the PAC foreign type code of the object with the leading '3' pared off (eg, '3JCL' would turn into 'JCL'). If [API-STFF-PARM.TYPE1] = 'N', then API-STFF-PARM.TYPE2 may carry the one character code of the NATURAL type of the object (eg, 'N' for subprogram or 'P' for program). If [API-STFF-PARM.TYPE1] = 'N' and [API-STFF-PARM.TYPE2] = ' ' and an suitable versioned object is found, then its Natural type code is returned in API-STFF-PARM.TYPE2.

[API-STFF-PARM.LINE-CNT] is made 0, if negative; it is made 10, if greater than 10. It is then interpreted as the number of lines (the size of the chunk) to be returned. In particular, no lines will

be returned, if LINE-CNT = 0. APINSTFF returns in API-STFF-PARM.LINE-CNT the number of lines of the returned chunk. It may be less than the adjusted original value, if the end of the DDM, loadable, source, USED list, or USED BY list has been reached.

API-STFF-PARM.LINE will carry the returned chunk.

API-STFF-PARM.OTHER-APPLICATION will be used if [API-STFF-PARM.TYPE3] is 'B' or 'U'. In this case API-STFF-PARM.OTHER-APPLICATION(I) will normally carry the name of the compartment holding the object referred to in API-STFF-PARM.LINE(I) for any I in {1,...,[API-STFF-PARM.LINE-CNT]}. A list can contain a non-object-reference line or a dead reference line; an API-STFF-PARM.OTHER-APPLICATION(I) may carry a compartment name if API-STFF-PARM.LINE(I) does not carry an object reference and it will be empty if [API-STFF-PARM.LINE(I)] is a dead reference.

Normally, API-STFF-PARM.API-PAC-AREA should be reset when the first chunk of a DDM, loadable, source, USED list, or USED BY list is wanted. API-STFF-PARM.API-PAC-AREA should be left untouched for a next call to obtain an immediately following chunk of the same DDM, loadable, source, USED list, or USED BY list. Set the last three bytes of API-STFF-PARM.API-PAC-AREA to H'F0F0F1' if the directory lines of the object are not to be skipped (otherwise they are skipped).

## API Return Codes

APINSTFF returns the following [API-PARM.API-MSG-NO]:

0000 if an object has been found and a non-final chunk is being returned.

1011 if the specified [API-PARM.API-MAIN-FUNC] is meaningless (ie, is not 'D').

6448 if [API-STFF-PARM.TYPE1] is meaningless (ie, is not 'D', 'E', 'F', or 'N').

6531 if the combination of [API-STFF-PARM.TYPE1] and [API-STFF-PARM.TYPE3] is inadmissible.

7205 if the "initialisation" has failed, ie, logical files 210 and 211 have not been found set at a pair of matching ACF and PCF.

7251 if the application has not been found or no object has been found.

9999 if an object has been found and a final chunk is being returned.

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.

# 37 APINSTT

API Routine: APINSTT
Parameters: UGNLPARM, UPILSTT

This API lets obtain data from a PAC status, create a new status, and modify an existing one.

APINAPP can be invoked with:

```
CALLNAT 'APINSTT' API-PARM API-STT-PARM
```

A definition of API-PARM is provided in LDA UGNLPARM in library SYSPACUS.

Set API-PARM.API-MAIN-FUNC to:

A To add a status.
D To obtain data from a status.
M To modify a status.

This chapter covers the following topics:

## Local Data Area UPILSTT

A definition of API-STT-PARM is provided in LDA UPILSTT in library SYSPACUS.

| Field Name | Format | Description |
|---|---|---|
| API-STT-PARM | | |
| STATUS-TYPE | A2 | in/out |
| STATUS | A32 | in |
| NOTES | A60 (1:3) | in/out |
| NEIGHBOUR-APPL | A32 | in/out |
| FUSER-DBNR | N5 | in/out |
| FUSER-FNR | N5 | in/out |
| FDIC-DBNR | N5 | in/out |
| FDIC-FNR | N5 | in/out |
| FPAA-DBNR | N5 | in/out |
| FPAA-FNR | N5 | in/out |
| XREF | A1 | in/out |
| ADD-TIME | T | out |
| ADD-BY | A8 | out |

| Field Name | Format | Description |
|---|---|---|
| ADD-TID | A8 | out |
| MOD-TIME | T | out |
| MOD-BY | A8 | out |
| MOD-TID | A8 | out |

Whenever APINSTT is called: API-STT-PARM.STATUS should carry the name of the status.

If [API-PARM.API-MAIN-FUNC] = 'D' and the status is found, APINSTT returns in the remaining parameter fields the status data, ignoring any values put by the user in the "in/out" fields.

If [API-PARM.API-MAIN-FUNC] = 'A' and the status is not found, APINSTT tries to add a NATURAL status with the characteristics specified in the "in/out" fields.

If [API-PARM.API-MAIN-FUNC] = 'M' and the status is found, APINSTT tries to modify the status endowing it with the characteristics specified in the "in/out" fields.

The meaningful values of the "in/out" fields are enumerated in the following table:

At application addition or modification user-specified values of the "in/out" fields may be translated.

Embedded spaces are replaced with '_' in user-specified values of PREFIX and LEVEL.

A meaningless value of PRD-APPL specified for the addition or modification of a Natural application causes the function to be abandoned. Other meaningless values of the "in/out" fields are treated as follows:

| Field Name | Format | Possible Values |
|---|---|---|
| STATUS-TYPE | A2 | A, D, I, M, N, P, R, T |
| NOTES | A60 (1:3) | Any |
| NEIGHBOUR-APPL | A32 | The name of a Natural or Predict application or blank. |
| FUSER-DBNR | N5 | [0..254,256..65534] |
| FUSER-FNR | N5 | [0..5000] |
| FDIC-DBNR | N5 | [0..254,256..65534] |
| FDIC-FNR | N5 | [0..5000] |
| FPAA-DBNR | N5 | [0..254,256..65534] |
| FPAA-FNR | N5 | [0..5000] |
| XREF | A1 | N, R, or blank |

With any of the three meaningful values of API-PARM.API-MAIN-FUNC the significance of the values of the API-STT-PARM fields is the same as that of their menu system counterparts.

At status addition or modification user-specified values of the "in/out" fields may be translated.

If NEIGHBOUR-APPL does not carry the name of a Natural or Predict application, it is reset.

A meaningless value of a -DBNR or -FNR field is translated to 0.

A meaningless or inappropriately specified value of XREF is translated to 'N' or space.

## API Return Codes

APINSTT returns the following [API-PARM.API-MSG-NO]:

1011 if an [API-PARM.API-MAIN-FUNC] distinct from 'A', 'D', and 'M' has been specified.

7201 if an ACF control record has been found defective.

7205 if the "initialisation" has failed, ie, logical files 210 and 211 have not been found set at a pair of matching ACF and PCF.

7250 if the status has been found.

7251 if an status has not been found.

7253 if the status has been added.

7254 if the status has been modified.

7257 if the status has not been added.

7259 if the status has not been modified.

7541 if the status (to be modified) is locked.

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.

# 38 APINTEXL

This API lets you obtain chunks of

- and migration lists of events containing URLs, `(API-FUNC = 'ME')`,
- migration lists of finalized archiving events `(API-FUNC = 'MH')`,
- audit reports of events `(API-FUNC = 'RE')`,
- audit reports of finalized archiving events `(API-FUNC = 'RH')`,
- descriptions of events `(API-FUNC = 'DM')`,
- descriptions of maintenance requests `(API-FUNC = 'DE')`,
- JCL texts `(API-FUNC = 'JJ')`.

APINTEXL can be invoked with:

```
CALLNAT 'APINTEXL' API-PARM API-TEXT-PARM
```

This chapter covers the following topics:

- **Local Data Area UGNLTEXL**
- **API Return Codes**

## Local Data Area UGNLTEXT

A definition of API-TEXT-PARM is provided in LDA UGNLTEXT.

| Field Name | Format | I/O |
|---|---|---|
| API-TEXT-PARM | A32 | in |
| OBJECT-NAME | A32 | in |
| LINE-CNT | P5 | in/out |
| TEXT-RECORD | A253 (1:10) | out |
| PAC-AREA | B20 | in/out |

When APINTEXT is called:

- API-TEXT-PARM.OBJECT-NAME should carry the name of the entity (event, finalized archiving event, maintenance request, or JCL text).
- [API-TEXT-PARM.LINE-CNT] is made 0, if negative; it is made 10, if greater than 10. It is then interpreted as the number of lines (the size of the chunk) to be returned. In particular, no lines will be returned, if LINE-CNT = 0.

- Normally, API-TEXT-PARM.PAC-AREA should be reset when the first chunk of the list, report, description, or text is required and left untouched for a next call to obtain an immediately following chunk of the same list, report, description, or text.
- APINTEXT returns in API-TEXT-PARM.LINE-CNT the number of lines (the size) of the returned chunk. It may be less than the adjusted original value if the end of the list, report, description, or text has been reached.

## API Return Codes

APINTEXL returns the following [API-PARM.API-MSG-NO]:

0000 if the entity has been found and a non-final chunk is being returned,

1011 if the specified ([API-PARM.API-MAIN-FUNC],[API-PARM.API-SUB-FUNC]) is meaningless,

7251 if the entity has not been found,

9999 if the entity has been found and a final chunk is being returned.

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.

# 39 APINTEXT

**Note:** Use **APINTEXL** instead.

This API lets you obtain chunks of

- and migration lists of events `(API-FUNC = 'ME')`,
- migration lists of finalized archiving events `(API-FUNC = 'MH')`,
- audit reports of events `(API-FUNC = 'RE')`,
- audit reports of finalized archiving events `(API-FUNC = 'RH')`,
- descriptions of events `(API-FUNC = 'DM')`,
- descriptions of maintenance requests `(API-FUNC = 'DE')`,
- JCL texts `(API-FUNC = 'JJ')`.

APINTEXT can be invoked with:

```
CALLNAT 'APINTEXT' API-PARM API-TEXT-PARM
```

This chapter covers the following topics:

- **Local Data Area UGNLTEXT**
- **API Return Codes**

## Local Data Area UGNLTEXT

A definition of API-TEXT-PARM is provided in LDA UGNLTEXT.

| Field Name | Format | I/O |
|---|---|---|
| API-TEXT-PARM | A32 | in |
| OBJECT-NAME | A32 | in |
| LINE-CNT | P5 | in/out |
| TEXT-RECORD | A80 (1:10) | out |
| PAC-AREA | B20 | in/out |

When APINTEXT is called:

- API-TEXT-PARM.OBJECT-NAME should carry the name of the entity (event, finalized archiving event, maintenance request, or JCL text).
- [API-TEXT-PARM.LINE-CNT] is made 0, if negative; it is made 10, if greater than 10. It is then interpreted as the number of lines (the size of the chunk) to be returned. In particular, no lines will be returned, if LINE-CNT = 0.

- Normally, API-TEXT-PARM.PAC-AREA should be reset when the first chunk of the list, report, description, or text is required and left untouched for a next call to obtain an immediately following chunk of the same list, report, description, or text.
- APINTEXT returns in API-TEXT-PARM.LINE-CNT the number of lines (the size) of the returned chunk. It may be less than the adjusted original value if the end of the list, report, description, or text has been reached.

## API Return Codes

APINTEXT returns the following [API-PARM.API-MSG-NO]:

0000 if the entity has been found and a non-final chunk is being returned,

1011 if the specified ([API-PARM.API-MAIN-FUNC],[API-PARM.API-SUB-FUNC]) is meaningless,

7251 if the entity has not been found,

9999 if the entity has been found and a final chunk is being returned.

Other values of API-PARM.API-MSG-NO may be received in more unusual situations.