

Object Types for the List

Objects fall into basic groups: Predict, foreign, Natural and user error message. DDMs and rules are recognized as Natural objects. Files and verifications are recognized as Predict objects.

This chapter covers the following topics:

- Predict Objects
- Foreign Objects
- Natural Objects
- Natural Dynamic Source Objects
- User Error Message Objects

Predict Objects

A Predict object is an entity in the Predict environment. For Predict objects, the reference in the format is optional:

object-name,object-type[,reference]

You may specify single or multiple-character codes for Predict object types as indicated in the following table; single-character codes are translated to the longer codes during migration list validation.

Code	Translates to . . .	Object Type
C	FIC	File type (e.g., DB2, views, conceptual files, standard files)
D	DA	Database
F	FILE	Adabas file
G	PA	Package list
K	KY	Keyword
L	RL	Relationship
N	DS	Data space
O	ST	Storage space
P	PR	Program
Q	VM	Virtual machine
R	VE	Verification
S	SY	System
U	US	User
V	VIEW	View
W	NW	Network
*		All objects

Note:

User-defined entities (UDEs) may not be specified.

Controlling the Predict Objects Included in a Migration

PAC uses the Predict SYSDICBE migration utility to load and unload Predict objects. Because of the nature of this migration utility, it is important to explicitly define the Predict object type. A single entry in the migration list may result in numerous Predict objects being processed. For example, if Predict type F or C is specified for an object, Predict automatically unloads the file together with all of the verifications and userviews linked to the file, and generates new automatic rules and DDMs for each object.

If you only need userviews and/or verifications for the migration list, place them individually in the list and omit the file itself.

However, the parent file (F or C) must have been previously migrated into PAC using the Predict migration utility SYSDICBE or an error will occur. Refer to Migrating Predict Objects, for detailed information. Refer to the Predict documentation for more information about the Predict migration utility.

Foreign Objects

A foreign object is an entity that exists outside the Natural system file. It may reside in a partitioned dataset (PDS) under OS/390 or in an LMS library under BS2000/OSD. Foreign objects can be source code or object code or both. They are loaded into PAC without being compiled or recompiled.

For foreign objects, the reference in the format is optional:

object-name,object-type[,reference]

Foreign objects in a migration list are identified by the prefix 3 (for example, 3COB is recognized as a COBOL object).

The PAC administrator determines the classes and types of foreign objects that are supported at your site and a project leader has the ability to further limit that set of classes and types for a particular application.

Examples of foreign object classes and types that might be supported at your site include:

- JCL
- APL Load
- Assembler Copycode
- Assembler Load
- Assembler Program
- Assembler Macro
- DBRM
- DBRM Load

Selecting Foreign Objects

Before foreign objects are eligible for a migration list, your PAC administrator must have defined their datasets to PAC and to the application.

Range notation is not currently available for foreign objects.

1. Enter FTYPES on the command line of the Migration List Editor to display a list of valid foreign object types.

PAC displays a list of all datasets defined for the application.

2. From the list of all datasets defined for the application, select the dataset.

A list of the contents of the selected dataset appears.

3. From the list of dataset contents, select the objects you want to be included in the migration list.

Restrictions

Foreign objects cannot be expanded. However, Natural objects can be expanded to include foreign objects as subordinates.

You can generate a list of foreign objects only where the origin (From) status is the Control status. Foreign objects cannot be archived.

Natural Objects

The format used for Natural objects in the migration list depends on the particular migration path being used for the migration event.

The following Natural object types may be specified:

A	Parameter data area	R	Report
C	Copy code	S	Subroutine
G	Global data area	T	Text
H	Helproutine	Y	Expert model
K	Server	Z	Recording
L	Local data area	*	All objects
M	Map	3	Dialog
N	Subprogram	4	Class
P	Program	5	Processor

Rules and Views

You may specify rules and views if they were generated when the object was identified to PAC by a Predict migration that generated the rule and/or view during the migration load. Object entries may also reference views/DDMs that were incorporated into PAC. Because these objects are already under the control of PAC, a status or version number has been assigned.

Note:

Other than the views incorporated into PAC during the initial setup of Predict information in PAC, views and rules may not be specified for incorporations.

If the same view or rule is specified multiple times in the migration list, only the last entry is used.

Migration into PAC

Natural objects being migrated online from development, maintenance, or incorporation status types, with or without a work file as input, are specified in the migration list using the following format:

$\left\{ \begin{array}{l} \text{object-name[,object-type]} \\ \text{object-type} \end{array} \right\} \left[\begin{array}{l} \text{,from-status} \\ \text{,version} \end{array} \right]$

You may use range notation for the object name and when specifying rules and views, even if you also specify the version number or status name.

From Development or Maintenance

When migrating objects from a development or a maintenance status, note the following:

- If the object name is specified, the type may optionally be specified, but if the type is inconsistent with the object name, a warning message may be issued indicating that the object type selected for the object is incorrect and that PAC has changed the type to be consistent with the object.
- You may limit the range of objects by entering a specific object type (currently limited to one type per entry) so that only objects of the specified type are selected during processing; or enter an asterisk (*) or blank for all types.
- You may specify a status or status alias for a maintenance status type; you may not specify a development status type because PAC does not track objects which are implicitly in development.
- You may specify a version number if the object is a subordinate and under the control of PAC.

If a version number is specified for certain object types (that is, copy code; map; global, local, and parameter data areas), PAC invokes rolling for that object type. Refer to *Using the Rolling Facility* for information about the rolling facility.

- If neither version nor status is specified, PAC compiles the objects with the latest version of the subordinate object. If a view and/or rule is not available, PAC uses the latest version for the application.

In the case of rules and views, the default version number may not necessarily be the highest existing version, but rather the highest version assigned (linked) to that application determined in a migration event where the From status is a development or maintenance type.

- When migrating objects from a development status, PAC processes the migration list and compares the date-time stamp of the object specified in the migration list with the date-time stamp of the latest version stored in PAC.
 - If the date-time stamps are identical and the object type is copy code or text, then the object is ignored; however, this may be overridden by User Exit 9.
 - For all other Natural objects (for example, program, subprogram, subroutine, data areas), PAC continues to process the object, even though the program has not been modified, because a copy code, or a view used by the program may have been changed.

From Incorporate

- When objects are incorporated from the Natural library directly (without using a work file), the migration list may be created as if it were a migration from development or maintenance (note the restriction on views and rules in the section *Rules and Views*).

- When objects are to be incorporated with input from a work file, the migration list consists of a single entry only as shown in the following examples:

Entry Loads all . . .

- *,* objects except user error messages.
- A*,* objects starting with "A" except user error messages.
- *,P program-type objects.
- S>,M maps with a name greater than "S".
- *,E user error messages.

Note:

User error message objects cannot be included in a migration list with other object types.

Migration from Control (Including Alignment)

When migrating Natural objects from the PAC Control status, version numbers or statuses may optionally be provided to identify the specific object(s) to be included during processing.

Status is Specified

If a status (or status alias) is specified, the following format is used for the object in the list:

object-name[,object-type] ,status

- The object name must be specified; however, range notation may be used.
- PAC accepts the object only if it has been migrated to the specified status. PAC determines the version number when the migration event is processed.
- The object type may optionally be specified.
- Rules and views may be specified if they have been registered to the application.
- For archiving, multiple objects with different versions may be added to the migration list.

Version is Specified

If a version number is specified, the following format is used for the object in the list:

object-name[,object-type] ,version

- The object name with the specified version number is processed regardless of the status(s) in which it exists (except the Archive status); it is then ignored and the user is notified.
- Object type is optional, but if the type is inconsistent with the object name, an error or warning message is issued.
- Range notation may be specified for rule and view objects only (except with alignment).
- Version number for rule and view objects must be valid for the application and not for the versions created during the migration of Predict data; that is, the rules and views must be registered to the application.

Neither Version nor Status is Specified

If neither version nor status is specified, the following format is used for the object in the list:

object-name[,object-type]

- The object name must be specified. Range notation may be used.
- The object type may optionally be specified.
- PAC defaults to the highest existing version number for the specified object, unless that version is archived.
- In the case of rules and views, the default version number may not necessarily be the highest existing version, but rather the highest version registered to that application.

Migration from Test or Production

When migrating Natural objects from test or production status types (that is, from a status other than Control, maintenance or development), the version number or status may optionally be provided to identify the specific object(s) to be processed. However, the reference will default to the name of the origin status.

Status is Specified

If status is specified, the following format is used for the object in the migration list:

object-name[,object-type], status

- The object name must be specified. Range notation may be used.
- The object type may optionally be specified.
- PAC accepts the object only if it has been migrated to the specified status. PAC determines the version number when the migration event is processed.
 - If the Control status is specified, the version number will always be the most recent one.
 - Status may not be used as a reference for the Archive and Retire statuses.
- In the case of rules and views, the object must be registered to the (foreign) application.

Version is Specified

If a version number is specified, the following format is used for the object in the list:

object-name[,object-type], version

- The object name with the specified version number is processed regardless of any status(s) in which it exists.
- If the destination status is Archive:
 - multiple versions of the same object may be specified.
 - asterisk (*) notation may be used for version to select the oldest as opposed to the most recent version.
- The object type may optionally be specified. If the object type is inconsistent with the object, an error or warning message is returned to the user.

- For rules and views, range notation may be specified.
- Version number for rules and views must be valid for the application and not for the versions created during the migration of Predict data; that is, they must be registered to the application.

Neither Version nor Status is Specified

If neither version nor status is specified, the following format is used for the object in the list:

object-name[,object-type]

- PAC defaults to the version of the object that is currently in the status from which the objects are being migrated. If the destination status is Archive, the default is the oldest version as opposed to the most recent version.
- PAC assumes that every object listed with no override version or status must exist in the origin (From) status of the migration event; if not, the object is ignored.
- For rules and views, range notation may be specified.

Natural Dynamic Source Objects

To migrate the source as well as the cataloged objects of copy code, maps, or programs from development, maintenance, or incorporation status types, manually enter the following codes in the migration list:

CS for source of copy code

MS for source of a map

PS for source of a program

Objects migrated into PAC with these designations are internally flagged to indicate that when they are migrated, the source will be migrated and objects will not be compiled. Refer to Migrations Between Control, Test, and Production for more information about dynamic source variables.

User Error Message Objects

The migration list also supports user error messages objects. User error messages are specified on the migration list in the following format:

error-number[,reference]

where error-number is the number assigned to the user error message preceded by E (for example, E1234); and reference is the version number or status of the user error message. The reference is optional; you may leave it blank.

Since the number of error messages used by any one application may vary from 1 to 9999, range notation may be used as follows:

from error number - to error number

For example: E100-E200,E

Since they cannot be compiled, user error messages do not need to reside in the Control status.

User error messages are migrated into and out of PAC. In a migration list, user error messages are treated similarly to Natural objects.

PAC validates user error messages in the migration list in the same way it validates Natural objects. User error messages must exist in the origin status.

User error messages consist of short and long error texts. For each error message, up to sixty languages for the same error message can exist. (The correlation of language codes to language text is defined in the ULANGTAB Natural macro.)