

# **Predict Application Control**

**Glossary**

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.

Copyright © 1990-2016 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

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

Table of Contents

Glossary ..... 1

    A ..... 1

    C ..... 2

    D ..... 4

    E ..... 4

    F ..... 5

    I ..... 5

    J ..... 5

    L ..... 6

    M ..... 6

    N ..... 6

    O ..... 7

    P ..... 7

    R ..... 8

    S ..... 9

    T ..... 10

    U ..... 10

    V ..... 10

    W ..... 11

    X ..... 11



## Glossary

---

This glossary lists PAC terminology in alphabetical order.

### A

---

|                       |  |
|-----------------------|--|
| ACF System File       | The ACF system file (application control file) stores the saved (source) and cataloged (executable) code for every version of every object in the PAC-controlled environment. The ACF also stores extended information about object versions, applications, statuses, migration paths, migration events, and audits. |
| Alias                 | An alternative name that a user assigns to a PAC entity. The user can reference and access the entity by its name or by an assigned alias.   |
| Alignment             | Alignment is a Control-to-Control migration. Alignment recompiles application objects so that they all reference the same versions of subordinate objects. It can be used to keep track of object versions and dependencies.   |
| API                   | See <a href="#">Application Program Interface</a> .  |
| Application           | A collection of objects configured to accomplish a specific data-processing task. A PAC application definition includes the name, level, and specifications for processing defaults. The set of component objects may change during the life-cycle.  |
| Application Environ   | The Natural library and system file, the Predict file and the dataset where application objects are stored, and the operating system, teleprocessing system, and database manager with which the objects are developed, tested, or used. Also called an operating environment.                                       |
| Application Integrity | In PAC, application integrity means that the correct version of each component object is implemented into production and that each object version uses the correct version of subordinate objects.   |

|                               |   |
|-------------------------------|---|
| Application Life-Cycle        | The cycle of development, testing, use, and maintenance of an application. In PAC, the application status links and migration paths defined for an application control movement through the application life-cycle.   |
| Application Program Interface | An application program interface (API) lets you call a PAC function from a user-written program without logging on to PAC and invoking the function from within PAC. APIs provide flexibility in accessing PAC information and interfacing with your existing systems.  |
| Application Status Link       | An application must be linked to a status before objects can be migrated to that status. An application status link specifies where the application's Natural, Predict and foreign objects are stored at the status. It also specifies a file translation table, if applicable, and whether Predict Xref data will be included in migrations to the status. |
| Applymod                      | Applymods allow you to customize the PAC system for your site by overriding PAC default settings. By default, applymods can be applied when an event is authorized, but the PAC administrator can control their use.  |
| Archive                       | The archive facility removes Natural objects (except views/DDMs and rules) from the PAC-controlled environment. It migrates objects from the Control status to the Archive status. The archived objects are unloaded to an archive file and removed from the ACF file. Archived objects can be restored to PAC.   |
| Audit Report                  | A PAC audit report provides information about PAC migrations. A PAA audit report provides information about objects in production statuses.   |
| Authorizer                    | A PAC migration event must be authorized by a valid user ID before it can be processed. Each migration-path definition includes the user IDs of authorizers for a migration event along that path. See also <a href="#">Configurator</a> .  |
| Auto-Expansion                | PAC can expand object lists automatically to include all subordinate objects. This function can be specified in the migration-path definition or invoked in the object-list editor.   |

## C

---

|                   |  |
|-------------------|--|
| Catalog Directory | The directory that shows the operating environment and size of a saved object. |
| Cataloged Object  | A compiled and executable Natural object.                                      |

|                        |   |
|------------------------|---|
| Change Control Log     | <p>The change control log provides the following information about migrations to and from maintenance statuses:</p> <ul style="list-style-type: none"><li>■ the object version</li><li>■ the destination library</li><li>■ the database number</li><li>■ the file number</li><li>■ the user ID and terminal ID of the person checking the object out or in</li><li>■ the date and time of the migration</li></ul>                             |
| Check-Out/Check-In     | <p>The PAC check-out/check-in facility allows you to monitor migrations to and from maintenance statuses. In addition, it automatically logs detailed information about the migrations, which can be viewed during the migration (through a user exit) or subsequently in change control logs. See <a href="#">Change Control Log</a>.</p>  |
| Compare                | <p>Utility delivered with PAC used to identify changes or differences between two objects or two lists of several objects.</p>  |
| Configurator           | <p>A PAC configurator is authorized to define applications, statuses, application status links, migration paths, and file translation tables. The PAC administrator specifies in the user profile whether the user may configure these entities.</p>  |
| Control Status         | <p>When an object is migrated from development or maintenance into the PAC-controlled environment, it is automatically linked to the Control status. When the object version is promoted to other statuses, it is copied from the Control status to the destination library. Physically, the Control status includes the PAC ACF and PCF system files. See also <a href="#">ACF System File</a> and <a href="#">PCF System File</a>.</p>      |
| Controlled Environment | <p>The operating environments that are protected by PAC and governed by site-specific procedures defined in PAC. Migrations (including logical promotions) of application objects are restricted by site-specific paths and authorizations. Objects can be changed only in defined development and maintenance locations; they are recompiled when returned to a test or production location. Object versions are tracked and controlled.</p> |
| Cross-Reference Data   | <p>There are two kinds of cross-reference data in PAC:</p> <p>Predict Xref data provides extensive information about objects, invocation methods, variables, data areas, files, and fields used by an</p>   |

object, as well as entry points or functions in the object that can be invoked by other objects. You can store and view the information in the Predict file associated with an application status link, in a remote location, or both.

PAC object usage data is similar to Predict Xref data except that it associates specific versions of the objects. PAC uses this data internally to track the version of an object that is referenced by versions of other objects. You can view the specific version of an object that is used by a specific version of another object.

## D

---

|                              |   |
|------------------------------|---|
| Data Definition Module (DDM) | A Natural view of a database file. DDMs can be generated from Predict views.                  |
| DBID                         | A database ID number (in PAC, the abbreviation DBnr is usually used).                         |
| DBnr                         | A database ID number.   |
| Deployment                   | A PAA-controlled application that is made up of application and status.                       |
| Destination Status           | In a PAC migration path, the status to which objects are migrated. Also called the To status. |

## E

---

|                                |  |
|--------------------------------|--|
| Executable Code                | Code that has been compiled and can be translated into machine-readable instructions.  |
| Expansion                      | Expansion lists all objects used by a specified object or all objects that use the specified object for compilation or execution. For example, the "Objects Used" expansion of a subroutine might list copy code, a local data area, and maps; "Used by" expansion might list several programs that use the subroutine. This function, which may be invoked from the object-list editor or automatically by PAC during a migration event, can be used to build object lists. |
| Extended Attribute Description | A text description of an application or migration event. It is entered and modified in a Natural editor window.  |
| Entity                         | A PAC structure such as a status or event.   |



---

|       |   |
|-------|---|
| Event | An action that changes an application's status in PAC. In almost all cases, a migration event is referred to. |
|-------|---|

## F

---

|                              |  |
|------------------------------|--|
| File Adjust Function         | The PAC administrator can use File Adjust to facilitate and control the renumbering of PAC files (FSEC, ACF, PCF).   |
| File Translation Table (FTT) | <p>When an object that was developed referencing one set of databases and files is executed referencing a different set, PAC can dynamically recompile the object using a file translation table. The FTT translates the database and file numbers from the development environment to the destination environment. It enables users to execute applications against different databases and files without changing and re-compiling the code.</p> <p>The FTT to be used in a migration to a particular status is specified in the application status link and invoked during a migration event.</p> |
| Fnr                          | File number.   |
| Foreign Object               | An object that is not of Software AG. For example COBOL modules or JCL texts.  |
| From Status                  | In a PAC migration path, the status from which objects are migrated. Also called the origin status.  |
| FTT                          | See <a href="#">File Translation Table</a> .   |
| FUSER File                   | The Natural system file where an application's objects are stored. Also called a user system file.   |

## I

---

|                                      |  |
|--------------------------------------|--|
| Implement                            | To put an application or object into production.   |
| Incorporation (Incorporation status) | The PAC Incorporation facility brings production objects into PAC without recompiling them. PAC incorporates the objects by migrating them from an incorporation status to the Control status and assigning them version numbers. Incorporation can be used only until a version of the object is migrated from a development or maintenance status. |

## J

**Job** A PAC job contains the JCL or JCS for PAC functions, such as migrations. A PAA job identifies the objects migrated and the backups created in a load into a PAA-controlled production environment.

---

## L

**Library** A location for programming objects.

**Load** The second step in a migration using a work file; it stores an object in a library or a group of data items in a database file. In PAA, a load is the process of storing objects in a production library and/or Predict file. The PAA load job specifies each object included in the load.

**Location** The physical whereabouts of a PAA controlled entity, made up of Natural library, DBID, FNR or dataset.

**Lock** When a PAC object or entity (for example, a migration event), is processed, PAC blocks access (locks) the entity until the processing is complete. When an object is migrated to or from a library, PAC or PAA locks the library until the migration is complete.

Locked Data: A PAC entity or item of data locked for processing.

---

## M

**Maintenance Request** A maintenance request can be defined and linked to a migration event. The maintenance request allows you to group related maintenance activities and track maintenance activities performed in response to the request. It can be used to link PAC to an external problem-tracking system.

**Migration** A logical migration is the promotion of objects to the next phase (status) in the life-cycle; it may or may not include a physical migration. A physical migration is the physical movement of code from one location to another.

**Migration Event** A migration event migrates objects along a predefined path and includes the object list, job, and specifications for executing the job. The migration event must be authorized by a valid user ID before it can be submitted for processing.

**Migration Path** A migration path defines the origin and destination statuses, authorizers, and processing defaults for a migration event.

---

## N

---

Natural Object                      A unit of Natural code (for example, a subroutine or map).

## O

---

Object                                  A unit of code, that PAC has under its control.

Object History                      PAC maintains a history of all the versions of an object in the PAC-controlled environment, including the dates and times they were migrated to each linked status.

Object List                            A list of all objects to be migrated.

Object Version                      An object with a version number assigned by PAC. Each time an object is migrated into PAC from a development or maintenance environment, PAC assigns it a new version number.

The PAC and PAA systems assign two independent version numbers to an object: one when the object enters the PAC-controlled environment (the Control version) and the other when the object enters the PAA-controlled production environment (the production version). The PAA system tracks the correspondence between these two numbers.

Also called a versioned object or PAC object.

Open Systems                        Distributed platform for Natural applications. PAC is now able to handle objects coming from this platform.

Operating Environment            The Natural library and system file and the Predict file where application objects are stored, and the operating system, teleprocessing system, and database manager with which the objects are developed, tested, or used. Also called an application environment.

Origin Status                        The status from which an object is migrated; also called the From status.

## P

---

PAA (Predict Application Audit)    The Software AG tool for auditing applications that have been placed in ("migrated to") production status.

PAA-Controlled Environment      The production environments that are protected and audited by PAA. See also [Production status](#).

PAA System File                    PAA audit data is stored in this reserved system file.

|                            |  |
|----------------------------|--|
| PAC-Controlled Environment | The operating environments that are protected by PAC and governed by site-specific procedures defined in PAC. Migrations (including logical promotions) of application objects are restricted by site-specific paths and authorizations. Objects can be changed only in defined development and maintenance locations; they are recompiled when returned to a test or production location. Object versions are tracked and controlled. |
| PAC Entity                 | A structure in the PAC system (for example, a status or migration event).  |
| PAC Object                 | An application object under PAC control. A PAC object has a version number. The object may be any type of Natural or Predict object or foreign object or a Natural user error message. Also called an object version or versioned object.  |
| PCF System File            | Natural objects are cataloged and Predict objects are generated in the PAC PCF system file. The PCF contains cross-reference data, keywords, and the latest version of Natural and Predict objects in the PAC-controlled environment.  |
| PF Key                     | Program function key.  |
| Predict File               | The system file (FDIC) where Predict objects and Xref data are stored.   |
| Predict Generation         | External objects, such as rules, data definition modules, databases, and files, can be generated from Predict definitions. The PAC administrator can modify the Predict generation defaults in PAC.  |
| Predict Object             | A unit of Predict information (for example, a userview).   |
| Production Status          | An end-user environment defined to PAC. Any environment defined as a production status is protected and audited by PAA.  |

## R

---

|                |  |
|----------------|--|
| Range Notation | Range notation enhances the search criteria for building a selection list of objects or PAC entities. A blank or asterisk (*) displays a list of all objects or entities, beginning with the first. A value followed immediately by an asterisk displays all items beginning with the specified value; for example, "ORDERS*" returns all items that begin with "ORDERS". A value followed immediately by (< or >) displays, respectively, all items that are greater than or equal to the specified value, or less than or equal to it. |
|----------------|--|

---

|               |   |
|---------------|---|
| Retire Status | A logical status used to purge obsolete objects. A migration to a retire status purges the object from the library in the origin status and removes the object's link with the origin status.   |
| Rolling       | When objects are compiled during migration, some or all of the related objects that are subordinate to the migrated object (maps, copycode, data areas, rules, and views) may be needed in their earlier versions for the compiling. Normally, PAC assumes that the latest version of these objects should be used. Specifying earlier versions of subordinate objects for compiling an object is called rolling. PAC provides a rolling facility for this purpose. |

## S

---

|                        |  |
|------------------------|--|
| Saved Directory        | The directory that shows the operating environment of a saved object.  |
| Saved Object           | A Natural, Predict or foreign source object that has not been compiled.  |
| Selective Processing   | This online facility lets you use search criteria to build a selection list of objects or instances of PAC entities (for example, statuses).   |
| Shared Table           | A file translation table that is shared with other application status links.   |
| Source Code            | Code written in a high-level language such as COBOL or Natural. It must be compiled and translated into executable code. See also <a href="#">Executable Code</a> .  |
| Status                 | Logically, a status is a stage or milestone in the life-cycle of an application. A PAC status definition specifies the type of environment (see <a href="#">Status Type</a> ), a default user system file, and a default Predict file.                             |
| Status Type            | PAC status types govern how PAC handles objects. There are currently nine types: Development, Test, Production, Maintenance, Control, Incorporation, Archive, and Retire.  |
| Step Library (Steplib) | A library linked to another library; when an object is requested from a library but not found there, the library's step libraries are searched.  |
| Substitution Parameter | Substitution parameters are variables in job control statements that allow you to substitute values, either automatically or by user input, at execution time. Substitution parameters are identified by an "@" prefix or a user-specified substitution character. |
| System Applymod        | See <a href="#">Applymod</a> .   |

|             |   |
|-------------|---|
| System File | <p>System files are Adabas files that contain software resources or user-created application objects. System files include the following:</p> <ul style="list-style-type: none"><li>■ FNAT contains system programs and utilities.</li><li>■ FDIC contains Predict objects and data.</li><li>■ FSEC contains security information.</li><li>■ FUSER contains application libraries.</li></ul> <p>The ACF, PCF, and PAA system files are reserved to PAC and PAA.</p> |
|-------------|---|

---

## T

|                   |  |
|-------------------|--|
| Textual Attribute | See <a href="#">Extended Attribute Description</a> .   |
| To Status         | In a PAC migration path, the status to which an application is migrated; also called the destination status. |

---

## U

|                  |  |
|------------------|--|
| Unload           | The first step in a migration using a work file; it moves or copies objects or data from a library or database file to a work file.  |
| User Environment | The mix of hardware, operating systems, teleprocessing systems, languages, and database managers.  |
| User Exit        | A user exit passes control to a user-written Natural program.  |
| User Profile     | The user profile, which is maintained by the PAC administrator, defines each user to PAC. It includes the user's name and ID, batch ID, and level of authority and access. |

---

## V

|                             |  |
|-----------------------------|--|
| Validation (of Object List) | When it validates an object list for a migration event, PAC ensures that all objects in the list exist in the origin status and that object names and types correspond.  |
| Versioned Object            | An object with a version number assigned by PAC. Each time an object is migrated into PAC from a development or maintenance environment, PAC assigns it a new version number. Also called an object version or PAC object. |
| View                        | A subset of the fields in a database file. Views can be documented in Predict. A Natural view, which is called a data definition module (DDM), can be generated from a Predict view.                                       |

## W

---

Work File                      A file used for temporary storage of objects or data.

## X

---

Xref Data                      See [Cross-Reference Data](#).

