



Chicago Interface Group, Inc.

CIG Package Utilities
Functional Overview

Chicago Interface Group, Inc.
858 West Armitage Avenue #286
Chicago, IL 60614 USA

Phone: (773) 524-0998
Fax: (815) 550-6088
Web: www.cigi.net
Email: support@cigi.net

CIG Package Utilities 7.0
Documentation version April 3, 2006

Copyright © 2007 Chicago Interface Group, Inc. All Rights Reserved
CIG Package Utilities is a trademark of Chicago Interface Group, Inc.
CA-ENDEVOR® is a registered trademark of Computer Associates International, Inc.

Introduction

What are CIG Package Utilities?

CIG Package Utilities (Package Utilities) help CA-Endevor (Endevor) package users work more efficiently and effectively. The product, a combination of exits and external utilities, provides an alternate path to Endevor package management. Package Utilities components, shown in figure 1 below, are as follows:

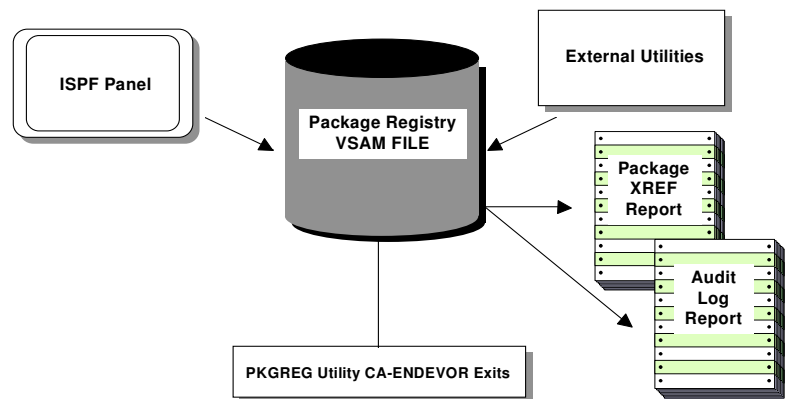
- back-end process installed in Endevor package exits allowing for management of element collisions, execution, and reuse of packages
- utility for printing, archiving, and clearing the audit log
- utility for reusing an executed package
- utility for building element and package cross-reference reports

- ISPF front-end for viewing the Package Utilities registry file and running reports.

Why Use Package Utilities if You Have Endevor?

Package Utilities enhances and simplifies Endevor package processing. You can now reuse an existing package, which reduces package build time, administrative overhead, and DASD usage. Packages can be rebuilt automatically, promoted to the next stage, or regenerated in place. You now have one central location which contains audit trail information listing the “who, what, and when” for each package. Finally, with four different options, you can decide what action to take if an element collision occurs on a CAST or a non-package action.

Figure 1 shows the structure of Package Utilities components. There are external utilities for reporting, package remake, and file maintenance; two reports, the audit log report and cross-reference report; exits; and ISPF panels for report generation and file listing



*Figure 1
Package Utilities structure*

What Happens to Existing, 'In Progress' Packages?

Package Utilities work with packages at any stage of the package life-cycle. The product is designed as a drop-in: no analysis, definition, or set-up is required. The first time any package is acted upon, the Package Utilities will determine if the package has ever been registered in the Package Utilities Registry File. If the answer is "no," then the package is registered at that time. For instance, if the user has a dozen packages that are "in-edit" status, the Package Utilities would intercept these packages as soon as a user modifies or casts the package. If the package is already approved, the Package Utilities would intercept the package during the 'before exec' exit point.

Is There a Way to Transfer Historical Packages?

For users who want to offload all historical packages, there is an ARCHIVE exit point that will perform a one-time package data extract and then build an audit log. Minimally, each package will have log information that reflects the major dates and users of the package. In addition, approvers and action data will also be transferred to the log as if the log was active at the time of approval and execution.

To invoke the transfer process, you must execute the PACKAGE ARCHIVE command against the packages you wish to have offloaded. There is no additional interface to learn or invoke.

Package Utilities will then automatically transfer Endeavor package information from the "before archive" exit point into the Package Utilities Registry. Once this data is transferred you can then perform PACKAGE DELETE command and remove historical information from Endeavor's package file.

Usage Scenario #1

Reusable Packages

The Need For Reusable Packages

The most common use of Endeavor packages is to promote a predefined list of elements via a package up the life cycle map.

Typically, programmers or managers will build a package at the entry point of the map and then request that the elements in the list be promoted up the map in a series of moves. With standard Endeavor packages, packages must be rebuilt at every stage.

Enter CIG Package Utilities . . .

After a Endeavor package is created, Package Utilities monitors every action against the package id. With Package Utilities, Endeavor packages can be reused. Automatically or via a batch utility, the old package will be remade, the element list will be promoted, and batch package SCL will be created to recycle the package id in Endeavor. All of this happens invisibly to the user and without losing historical package information. For example, see figures 2 and 3.

Figure 2: Without Package Utilities, a programmer or administrator must rebuild the package from a fresh element list at every stage. Because of audit trail requirements, the package is often built with a new name to retain the old package information.

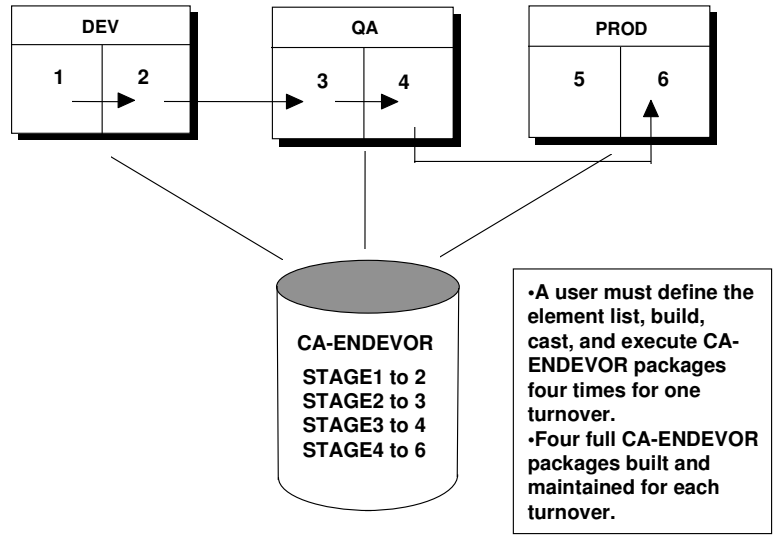


Figure 2 An example of one turnover without CIG Package Utilities

Figure 3: With Package Utilities, the programmer has to build the element list only once, and the list will be moved up the map automatically through the REMAKE facility.

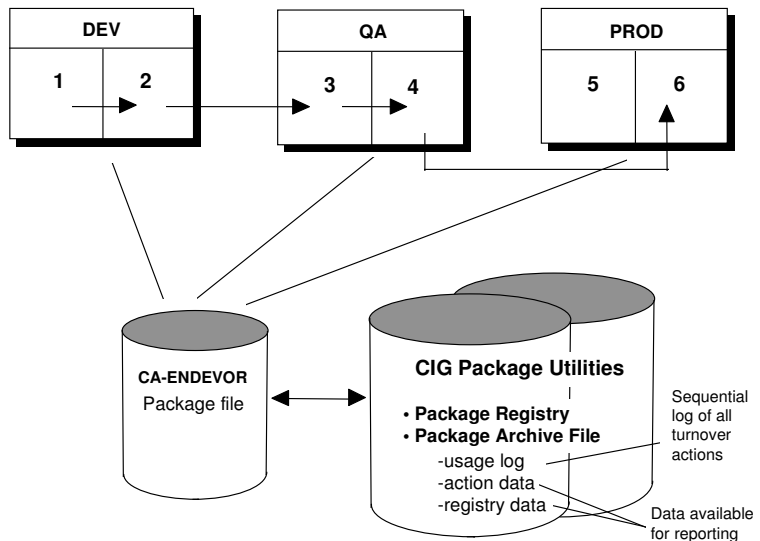


Figure 3 An example of one turnover with Package Utilities

Usage Scenario #2
 Element Collision Management

Element Collisions Due to Parallel Development

In the best of all worlds, only a single programmer would work on a single piece of code at one time. All updating would be at a controlled pace and would be sequential in nature. In contrast, most Endeavor users are faced with ever-increasing demands for systems changes, both long-term and short-term. In many shops, the same module may

be updated by different people, teams, or locations. When not managed, this parallel development activity leads to code regressions and collisions.

With standard Endeavor packages, an element is CAST into a package but not frozen from use by other packages or subsequent Endeavor actions. Even after being CAST, the element can be retrieved, updated, deleted, CAST into another package, or even moved by another package. This element activity results in package failures and incorrect code implementation. For example, see figure 4.

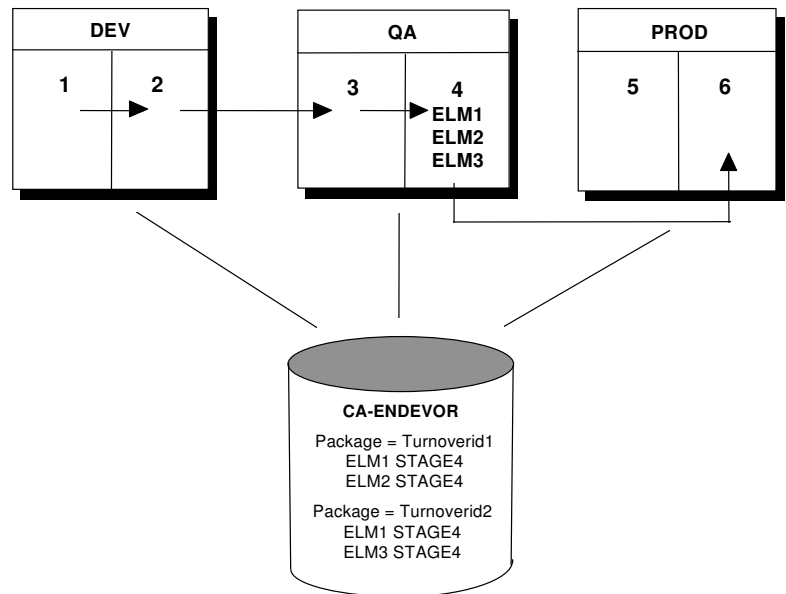


Figure 4 shows a case where ELM1 has been CAST into two different packages. In the event ELM1 is subsequently modified, package failure will occur and the incorrect level of element ELM1 will be promoted.

*Figure 4
 Same element attached to multiple packages*

Enter CIG Package Utilities .

. .

At Endeavor CAST time, each element is checked against an element registry of existing packages. Assuming the package has no collisions, the package is registered in the Package Utilities Registry. If the element is found to be CAST to another package id, then one of four events occur based on user setup options:

- package is failed
- package status is set to “resolve” with all other packages involved in the collision
- the user is warned of a collision
- collision is ignored.

In the example shown in figure 5, Package Turnoverid1 was CAST first and its elements were registered. When Turnoverid2 was CAST, there was an element collision with Turnoverid1. The package was fully CAST, but set to RESOLVE status. Turnoverid1 was also set to RESOLVE status. Prior to either of these packages being executed or approved, the packages would need to be RESET and the collision would need to be resolved. The log of each package id would be updated to reflect collisions, warnings, and resolution.

Figure 5: The Package Utilities Registry is used to manage element collisions. At Package CAST, the registry is checked for element collision situations. Collision management is controlled via user-specified collision rules.

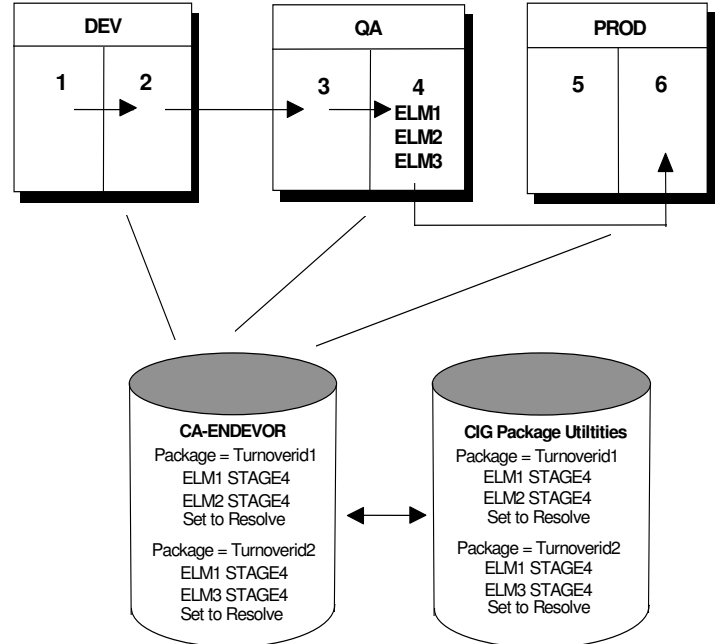


Figure 5
CAST collisions checks and events

Usage Scenario #3
 Audit Trail &
 Minimizing DASD Usage

*Full History of Package id
 Activity*

Most Endeavor users have very serious auditing requirements dictating that an audit trail of all element activity be kept. Traditional Endeavor package processing builds many records per package id. If the package id is used again, then all history is lost, and thus the audit trail is lost. For audit reasons, most shops keep fully executed package ids for many months. Consequently,

the package file continually increases in size, resulting in significant performance degradation and excessive DASD utilization.

Enter Package Utilities . . .

Every time the package id is updated, an audit log record is recorded. These log entries remain as historical record even if a package is RESET, modified, deleted, or CAST. This audit trail can also be extracted for additional reporting and auditing activities. For example, see figure 6.

Figure 6 shows an example of a log for a package that has been built, executed, and processed via the REMAKE action. In this example, two elements have been moved twice from stage A and then from stage B.

ACTIVITY	USER	DATE	TIME	PC
Date 95/06/25 TIME16:59 PACKAGE UTILITY AUDIT LOG REPORT				
FOR PACKAGE: TEST1 STATUS: APPROVED UTILIT				
CREATE	CIG01A	95/06/24	12:57	00
CAST	CIG01A	95/06/24	12:57	00
MOVE	\$CPOOL (01.00)	TEST	SYSA	
MOVE	\$ENTRY (01.00)	TEST	SYSA	
EXECUTE	CIG01A	95/06/24	13:01	04
REMAKE	TPSXXX	95/06/24	13:01	00
COMMIT	TPSXX	95/06/24	13:02	00
DELETE	TPSXXX	95/06/24	13:02	00
CREATE	TPSXXX	95/06/24	13:02	00
CAST	TPSXXX	95/06/24	13:02	00
MOVE	\$CPOOL (01.00)	TEST	SYSA	
MOVE	\$ENTRY (01.00)	TEST	SYSA	
EXECUTE	CIG01A	95/06/24	13:06	00
REMAKE	CIG02A	95/06/24	13:06	00
16:59 PKG3163I PRINTLOG COMPLETED SUCCESS				

*Figure 6
 Audit trail for Package 'Test 1'*

Immediate Offload of History and Reduction of Package File Size

Once you begin using Package Utilities you may want to transfer your current historical package ids into the Package Utilities Registry File. This allows Endeavor administrators to delete historical packages from the Endeavor package file, thus limiting the contents of the Endeavor package file to only active packages.

After running an ARCHIVE action against the package file, a PACKAGE DELETE action can be used to remove historical Endeavor packages from the Endeavor package file.

The result: a significant reduction in size of the Endeavor package file, along with improved performance.

Getting Started

Package Utilities is a “drop-in” product. Installation is quick and easy. The Package Utilities QuickLoad program allows existing Endeavor packages to be loaded into the CIG Package Utilities Registry. With QuickLoad you can immediately begin reporting on packages in your package file.

Package Utilities requires a minimum of Endeavor 3.9.

For more information on CIG Package Utilities or any other products or services available from Chicago Interface Group, call (773) 524-0998 option 1.

