



# Edge Portfolio Analyzer

## General Information

January 2012

V1 R07.03

Edge Portfolio Analyzer  
General Information

**The descriptions contained in this document represent the Edge Portfolio Analyzer functions as of the date shown, and are subject to change.**

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

IEEE is a trademark of the Institute of Electrical and Electronics Engineers, Inc. in the United States and other countries.

POSIX is a trademark of the Institute of Electrical and Electronics Engineers, Inc. in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

© Copyright 1994 - 2012 MCG, Edge Information Group. All rights reserved.

Printed from File: GIM-173X.WPD on March 4, 2012.



## Contents

<b>Chapter 1: Introduction to Portfolio Analysis</b> .....	1
Enterprise Management Perspective .....	4
Application Portfolio Management .....	4
Program Management .....	5
Keeping It All Together .....	6
<b>Chapter 2: Edge Portfolio Analyzer Overview</b> .....	9
Analyzer Summary Reporting .....	10
Analyzer Detail Reporting .....	11
Analyzer Derivative Reporting .....	13
External Interfaces .....	14
Automated Relink .....	15
External Tool Population .....	16
External Analyzers .....	16
Information in Detail Data Files .....	22
<b>Chapter 3: Analyzer Reports</b> .....	23
Sample Detail Reports .....	24
Sample Summary Reports .....	52
Sample External Analyzer Reports .....	59
<b>Chapter 4: Operating Environment</b> .....	65
<b>Appendix A: User Feedback</b> .....	A-1

## Figures and Tables

<b>Figure 1:</b> Levels of Portfolio Management .....	4
<b>Figure 2:</b> External Tool Interfaces .....	15
<b>Table 1:</b> Information in Detail Data Files .....	22

## Sample Reports

Sample Report 1: OS/VS COBOL Detail . . . . .	24
Sample Report 2: OS/VS COBOL NORES Library Levels . . . . .	25
Sample Report 3: VS COBOL II Detail . . . . .	26
Sample Report 4: COBOL/370 R1.x Detail . . . . .	27
Sample Report 5: COBOL for MVS and VM Detail . . . . .	28
Sample Report 6: COBOL for OS/390 and VM Detail . . . . .	29
Sample Report 7: Enterprise COBOL for z/OS and OS/390 Detail . . . . .	30
Sample Report 8: PL/I Version 1 Detail . . . . .	31
Sample Report 9: PL/I Version 2 Detail . . . . .	32
Sample Report 10: PL/I Version 2 - Fetch Example Detail . . . . .	33
Sample Report 11: PL/I for MVS and VM Detail . . . . .	34
Sample Report 12: Enterprise PL/I for z/OS and OS/390Detail . . . . .	35
Sample Report 13: Enterprise PL/I for z/OS V4.2 with Compiler Option Reporting . . . . .	36
Sample Report 14: C/370 Version 1 Detail . . . . .	37
Sample Report 15: AD Cycle C/370 Version 1 Release 2 Detail . . . . .	38
Sample Report 16: C/C++ for MVS/ESA Version 3 Release 1 Detail . . . . .	39
Sample Report 17: C/C++ for MVS/ESA Version 3 Release 2 Detail . . . . .	40
Sample Report 18: C/C++ for OS/390 Version 1 Release 3 Detail . . . . .	41
Sample Report 19: C/C++ for z/OS Version 1 Release 13 Detail . . . . .	42
Sample Report 20: FORTRAN G/G1 Detail . . . . .	43
Sample Report 21: FORTRAN H/H Extended Detail . . . . .	44
Sample Report 22: VS FORTRAN Version 2 Release 6 Detail . . . . .	45
Sample Report 23: System Date Requester Detail . . . . .	46
Sample Report 24: Alias and SETSSI Reporting . . . . .	47
Sample Report 25: Multi-Language Application Detail . . . . .	48
Sample Report 26: CEEDOPT - LE V1.5 Default Run-Time Options . . . . .	49
Sample Report 27: CEEDOPT - LE from OS/390 2.5 Default Run-Time Options . . . . .	50
Sample Report 28: CEEUOPT - User Specified LE Run-Time Options . . . . .	51
Sample Report 29: Summary Overview . . . . .	52
Sample Report 30: COBOL Compiler Related Statistics Summary . . . . .	53
Sample Report 31: COBOL Run-Time Components Statistics Summary . . . . .	54
Sample Report 32: PL/I Compiler Related Statistics Summary . . . . .	55
Sample Report 33: PL/I Bootstrap and Run-Time Statistics Summary . . . . .	56
Sample Report 34: Fortran and Other Languages Statistics Summary . . . . .	57
Sample Report 35: Other Entity Statistics Summary . . . . .	58
Sample Report 36: System Date Requests External Analyzer . . . . .	59
Sample Report 37: System Date Request Summary . . . . .	60
Sample Report 38: CSECT Consistency Report, COBOL RES/NORES Conflict Report . . . . .	61
Sample Report 39: PLESCNx: PL/I LE Migration Report . . . . .	62
Sample Report 40: CICSCBx: CICS V3.1 Migration Issues: OS/VS COBOL . . . . .	63

## Chapter 1: Introduction to Portfolio Analysis

Business needs require that existing applications, sometimes decades old, continue to operate in an evolving environment with a minimum of risk and maximum quality. Applications can outlive processors, operating systems, compilers and language implementations.

As IBM tightens the minimum operating environment for new releases of z/OS and various sub-systems such as CICS and DB2, it is becoming more and more critical to keep up, rather than constantly playing catch up. In some cases, the migration path requires that each step be taken in sequence without skipping a level.

A key issue in managing a large portfolio of existing applications is understanding the components of the portfolio and the relationships among those components. Successful migration to newer platforms and new middleware releases requires significant amounts of application understanding. Questions such as: "What is the makeup of an application?" "What technologies were used to implement the application?" "How widespread is the usage a common routine?" are vital to understanding the application.

In the past, many of these platform, operating system and middleware updates were as simple as an install of the new product level. Today the level of complexity and the corequisite requirements often dictate other migrations in order to accomplish the real objective.

For example, CICS TS V3 no longer provides for the execution of transactions written in OS/VS COBOL. Thus, if an application that is targeted for the new CICS is still compiled with the old compiler, a language and compiler migration must be performed. More so, if all of the applications are not using the Language Environment (LE) run-time support, that migration must also be performed before the CICS upgrade can be attempted. CICS V4 places even more restrictions on the application levels that are indicated as *supported* development environments. The minimum supported HLL compilers for CICS TS V4.1 are listed as:

- Enterprise COBOL for z/OS V4.1
- Enterprise COBOL for z/OS V3.4
- Enterprise PL/I for z/OS V3.5 or later
- C/C++ z/OS V1.8, or later

Of course, all of these require Language Environment for run-time support.

DB2 V8 has even more stringent requirements to exploit the new functions provided by that version. Applications using those new functions must be compiled with the latest levels of the Enterprise

Edge Portfolio Analyzer  
General Information

COBOL and/or Enterprise PL/I compilers. The IBM supported migration path to DB2 V9 requires a prior migration to DB2 V8.

Even the exploitation of new hardware or operating system capabilities may require upgrades to other components. Therefore, the days of *occasional migrations* is past. Migrations that require a more extensive understanding of the application portfolio are now the rule rather than the exception. A comprehensive inventory is required to make these evaluations to avoid undesirable surprises.

In the case of migration to, for example, zEnterprise processors, there are a large number of new instructions that can be key to obtaining all of the performance improvements inherent in the new hardware. But, how do you know if your application is exploiting those new instructions? The Enterprise PL/I and the XL C/C++ compilers control the exploitation of new instruction through the use of the ARCH(n) compiler option. (C/C++ also has a TUNE(n) option.) But your key to full exploitation of the new instruction is knowing what ARCH level your applications use. Over time, you may have a wide variation of levels. For an analysis of time-critical components to assure the best performance, you will want to concentrate on those components that have not kept up.

Upgrade projects require some specific and, in many cases, unique views of the application and the technologies used. And, as was proven in most Year 2000 projects and now in CICS and DB2 upgrade projects, a tool that can automate the analysis process is a valuable asset that can significantly improve the productivity of the management and project teams.

If you do not know what you have in your inventory, you do not know what you need to upgrade. Even when you know the components involved, there are often unique issues with testing and problem determination that require a special understanding of the functions used in a program. Just one example from Y2K involved programs that requested the current date from the system. You had to know how the request was made. And it was often desirable to know how the request was made, because this might dictate the methods and tools used for testing the program. Knowledge of date requests are still valuable, for example, when changing the base date of a system from local time to GMT/UTC time. Many installations are making this change as their business expands to be more of a world-wide operation. While the Y2K problem is, hopefully, all behind us now, the need to understand similar issues remains. Use of some system functions such as LINK or LOAD by Assembler routines can cause unpredictable behavior when these functions are moved to LE. Without a good tool to find the use of these functions, either the risks are higher or a manual effort may be required.

## Edge Portfolio Analyzer General Information

Migrations, whether Language Environment, a new release of the operating system, or a major sub-system upgrade are now a normal part of an IT operation. Any major system, middleware or application upgrade requires proper migration plans and testing plans that utilize many of the same basic types of information as were required for Year 2000. Another example is the IBM Statement of Direction (SOD) removing support for user modified CEEDOPTs to designate LE default run-time options after z/OS V1.13. Any changes to CEEDOPTs must be reflected in PARMLIB to be supported in the future.

Portfolio analysis is not limited to migration or application upgrades. It is part of the ongoing process of maintaining an up-to-date operating environment. Regular operating system upgrades have become more common as the demands for newer and more complex technologies increase. The technologies are changing so rapidly that the need to stay current is even more important than in the past. In addition, day-to-day functions such as problem source identification and problem determination have become more complex and require a detail understanding of the constructs of an application.

Language Environment (LE), for example, contains the run-time components that were previously part of each language product. And, LE is now tightly integrated into the z/OS operating systems. So, when you upgrade from one release of z/OS to the next, you are upgrading LE at the same time. It is even more important in this new environment that applications personnel and systems personnel work together to plan changes in the platform. Operations personnel and internal technical support groups will also have to be a part of the planning and implementation processes. Portfolio analysis can provide a common source of information for everyone involved in these ever-changing environments.

Effective portfolio analysis encompasses multiple levels of information. The scope and amount of detail must be consistent with the needs of those who are using it.

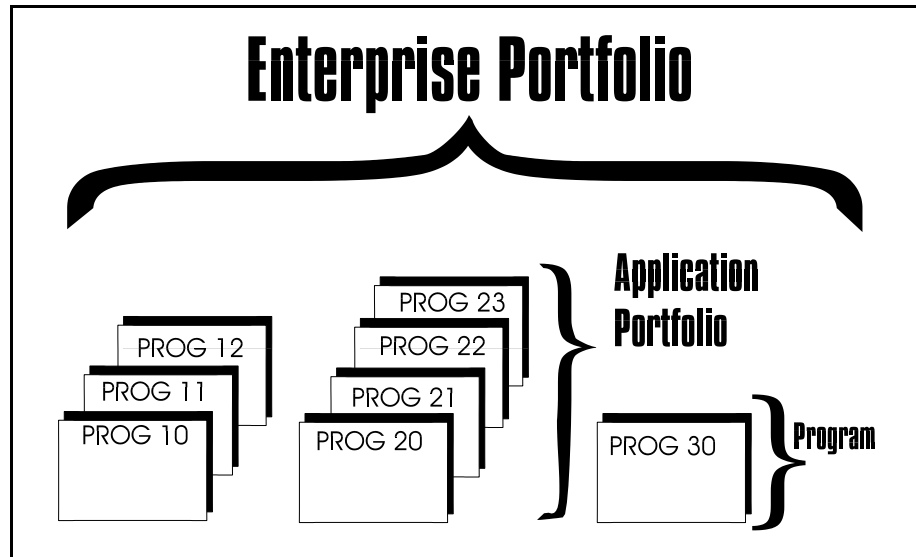


Figure 1: Levels of Portfolio Management

## ***Enterprise Management Perspective***

Effective management requires a big picture view of the portfolio. The concerns at this level are with the entire portfolio.

- How many programs are there?
- What languages and technologies are employed?
- Are appropriate quality standards being met?
- How fast is migration to a new technology taking place?

This perspective can be gained only by taking a look at the entire portfolio. From this perspective, managers can plan the most effective way to migrate to a new technology or implement a set of standards. With adequate information, they are able to focus limited manpower where it is needed most.

## ***Application Portfolio Management***

At the project level, portfolio management looks at a single application or possibly a family of programs. The goal is to identify programs that require special attention or handling. Questions might be asked such as:

- What programs will be affected by a change to a subroutine?
- What programs are dependent on specific vendor run-time routine?

- Which programs are "easy to migrate" and which contain specific migration inhibitors?

Project leaders can be most productive by understanding the complete application portfolio. They can avoid many otherwise unexpected situations that often arise when migrating or implementing other changes to existing applications. This level is also appropriate for operations planners who must assure that the appropriate levels of documentation and support are provided as a migration is implemented.

## ***Program Management***

Developers migrating, modifying or enhancing particular programs also need portfolio information.

- What components make up a particular program? And, are those components also used in some other application?
- What level of the compiler was used for each component? Will the compiler need to be changed as part of the application update?
- What compiler options were specified when the program was last compiled? In some cases, changing compiler options can change the results produced by a program. Maintaining consistent compiler options across the installation should be part of any configuration management plan. When an application is being changed, this is often the best time to introduce consistency since the application must be tested anyhow.
- Did the source management tool leave an "eye-catcher" in the program to help identify the program or changes made to the program?
- Which compiler run-time routines and run-time option overrides are included as part of the application load module?
- What is the release level of the run-time routines used or link edited into the application? Do the routines need to be refreshed to the latest level to migrate the application?
- During Y2K, it was typical to ask: Does the COBOL module use ACCEPT FROM DATE, or the PL/I module use DATE or DATETIME Builtin Functions to request the system date? Does an Assembler routine contain references to the system date by uses of the TIME system macro or the Store Clock (STCK) hardware instruction? But, even now, as operations become more international in scope, that

## Edge Portfolio Analyzer General Information

same information is used as systems are changed from using the local date/time as the base system date/time to using GMT or UTC as the standard base. Similar questions might be asked about the use of SORT, DB2, IMS, CICS or other LE or system functions if those components or subsystems are being upgraded. Some of the work being done for SOA modernization of applications has also exposed issues that may become problems with the so-called Y2038 and/or Y2042 problems. This is because the quick fixes often had time periods associated with them, and those clip dates are now much closer to current dates.

- Does the application make use of some Language Environment Callable Service that has changed in the latest release of LE? This happened in OS/390 V2.9 and V2.10.
- Does a COBOL program use dynamic CALLs to load an independently linked module for execution? Does a PL/I program use the FETCH/RELEASE statements for dynamic loading of routines?
- Does your inventory of CICS applications include PL/I and/or C/C++ applications that exploit the Additional Floating Point (AFP) registers in the newer System z processors? If so, do the programs have the necessary maintenance and/or are they using the required compiler options to protect against possible data corruption in this environment?
- Does an Assembler routine issue a LINK or LOAD macro to dynamically invoke a separately link edited module? And, is it the primary entry point of the module, thus making it more of an issue for migration.
- Do Assembler routines use other SVCs that may require attention when this application is modified?
- Are there some components in a load module that require updating or relinking to allow the program to run with LE or to run above the 16MB line?

Having the right information readily available makes the developer more productive. Problem areas can be addressed immediately, avoiding a lengthy trial-and-error process.

### ***Keeping It All Together***

The traditional method of portfolio management has been the maintenance of printed output. But, to be useful, you must have the



## Edge Portfolio Analyzer General Information

compile listings, associated JCL and linkage editor output and they must be kept up to date to reflect all changes made to the application, no matter who made the change and when the change was made. More important, the information must be readily available, even to the developer who receives a 3:00 AM call at home.

Inventories and analyses of source libraries provide some facts about the portfolio. But, in many cases, only limited information can be gained from source libraries. Source modules do not provide complete information about the technologies used, such as compiler and linkage editor options.

More important, source libraries are often incomplete and out of sync with the production libraries. This is especially true for seldom used programs such as those run annually or on demand. Likewise, purchased applications are often distributed in object or binary form. Source modules may not be available for these applications. Often, the load module is the only source of information that can be used to identify the origin of the program, or may even be the only indication that the program exists.

While compiler and link editor listings can provide some additional information, they are not always kept up to date. Without careful checking, it is never certain that the listing actually matches the code that is being executed.

Missing source programs are of particular concern when attempting to gain control of the application development process. But, it is not enough to know that there is a source program that matches an executable module in a load library. It is critical to know if there are multiple versions of that executable program, compiled on different dates, with different sizes, when there is only one copy of the source.

Production load libraries may be the only source for accurate information crucial to understanding your application portfolio. It is these modules, in fact, that contain the true information about the code that is running in the system.

Edge Portfolio Analyzer  
General Information



## Chapter 2: Edge Portfolio Analyzer Overview

The Edge Portfolio Analyzer is a proprietary utility that analyzes existing load libraries. It produces printed reports and **machine friendly** flat files reflecting the results of the analysis. Reports can be produced for the entire library or generic selection can be done to include or exclude specific modules or groups of modules.

Using the Edge Portfolio Analyzer (a.k.a., EPA), you can improve the performance of your production environment as well as reduce the exposure to problems when migrating to new compilers, new run-time environments and/or new operating system releases. At the same time, you can improve the productivity of the personnel who are charged with the responsibility of managing this environment.

The Edge Portfolio Analyzer helps you:

- Identify modules that contain components that will need attention when a new compiler release is installed, an update to z/OS being planned or a new version of sub-systems such as CICS or DB2 are installed,
- Identify load modules that are more likely to be easy to successfully migrate to a new compiler or other new environment.
- Identify load modules that will create difficulties during the migration to a new compiler and/or run-time environment.
- Improve performance of certain load modules simply by recompiling them with different options to take advantage of new hardware.
- Identify load modules containing Assembler CSECTs that use certain system functions via an SVC or Service CALL that may have changed in a new release, or that reference the system date using the TIME system macro and/or the Store Clock (STCK) hardware instruction. In addition, by using an External Analyzer provided as part of the package, high level language (HLL) program use of date facilities can be discovered.
- Locate load modules that will be impacted by changing a specified CSECT. (e.g., The date module mentioned above.)
- Determine the language used by all programs that use a particular facility of the system. (e.g., What languages are used to access DB2 facilities?)
- Ensure that production programs conform to the installation standards for the compiler and linkage editor options.
- Extract information that can assist in audits required to meet business and/or governmental standards. e.g., Has a critical component or module in an application been changed. And, if so, exactly when was the change made.

The Edge Portfolio Analyzer is a knowledge and experienced based "power tool" that eliminates the need to understand the obscure contents of load modules. Because of its design, it allows results in minutes instead of hours or even days required with basic system utilities and manual analysis.

## ***Analyzer Summary Reporting***

The Edge Portfolio Analyzer can produce summary information on the contents of an entire load library or by using generic name selection. This summary information is an inventory of all modules showing totals of modules that fit certain characteristics. The Summary Report contains statistical information such as:

- Overall library (or for selected members) contents
- Linkage Editor "troublemakers"<sup>1</sup>
  - "Not Executable" modules
  - "Overlay" modules
- For COBOL modules
  - Inventory of modules by compiler and version
  - Inventory of selected compilation attributes (options)
  - Inventory of run-time components for NORES programs
- For PL/I modules
  - Inventory of modules by compiler and version
  - Inventory of run-time components by version
- For FORTRAN modules
  - Inventory of modules by compiler and version
  - Inventory of run-time components by version
- For C Language modules
  - Inventory of modules by compiler and version
  - Inventory of selected compilation attributes (Compiler options are provided for C/C++ starting with the compiler from z/OS V1.10.)
  - Inventory of run-time components by version
- Inventory of Assembler generated modules
- Inventory of Language Environment<sup>2</sup> run-time components

---

<sup>1</sup> Detail Data Records are not created for Not Executable modules. Incomplete Detail Data Records are created for Overlay modules, but these modules can be fully analyzed and Detail Data Records created by relinking them without the OVERLAY option.

<sup>2</sup> The terms *Language Environment* and *LE* refer to the common run-time support routines for most IBM High Level Languages (HLLs). Beginning with Language Environment for OS/390 and VM the functions are now integrated into OS/390 and z/OS. LE may also be referred to as LE/370 or LE for MVS and VM by earlier releases of the product.

Edge Portfolio Analyzer  
General Information

- Inventory of Language Environment run-time option override components
- Inventory of VS Pascal, REXX, Easytrieve or other generated components
- Inventory of subsystem interface modules for IMS, CICS and DB2
- Inventory of "User Defined" Assembler components (e.g., Telon run-time, IEF run-time, ABEND, data base access)

This summary information is vital for gaining an overall understanding of your portfolio and for quickly highlighting troublemakers.

## ***Analyzer Detail Reporting***

Detailed information about individual load modules is also furnished by the Edge Portfolio Analyzer. The Detail Report describes the characteristics of each load module and of the CSECTs contained in the load module..

The following information can be determined from the detail report and/or the Detail Data File:

- Load module attributes
  - Date (and in most cases the time) of most recent linkage edit
  - Linkage Editor assigned attributes (RMODE, AMODE, NOT-EXEC, REENTRANT, REUSABLE)
  - Storage required to load the module
  - The value of the SSI field recorded by linkage editor or a SUPERZAP SETSSI control statement
  - Alias names are identified. In addition, *False Alias* or *Broken Alias* Chains can be diagnosed.
  - Use of MIXRES if set using IGZEOPD for VS COBOL II Release 3.x or Release 4 programs.
- Component (CSECT) attributes
  - Language and compiler version used for each component
  - Date and (when available) time of compilation
  - Size of the CSECT
  - RMODE/AMODE for the CSECT
  - For LE conforming applications, the size of the initial STACK required by the program.
  - The application uses the integrated preprocessor built into the COBOL and/or PL/I compilers for CICS and/or DB2 rather than the separate utilities.
- Additional COBOL Specific Information
  - Load module level information
    - Symbol at primary entry points
  - CSECT level information
    - Significant compiler options identified

Edge Portfolio Analyzer  
General Information

- Differences in versions of identically named VS COBOL II Bootstrap modules (IGZEBST)
- The distribution product for identically named OS/VS COBOL library modules (ILBOxxxx)
- Dynamic CALL usage
- ACCEPT FROM DATE, DAY or TIME usage
- "Eye-catchers" that have been placed in OS/VS COBOL Working Storage to further identify the version of the source that produced the object program.
- Additional PL/I Specific Information
  - Load module level information
    - Symbol at primary entry points
    - Fast Initialization usage
    - PL/I Shared Library usage
    - Linkage editor attributes
  - CSECT level information
    - Significant compiler options identified <sup>3</sup>
    - If the SERVICE compiler option is used to provide further user supplied version or application specific information, this information is extracted from the CSECT.
    - PL/I run-time default override module identified and decoded
    - Does the generated code take advantage of the performance improvements provided by new hardware instruction. (i.e., What is the ARCH compiler option value?)
    - PL/I run-time library components identified by version and release
    - Size of the Dynamic Storage Area (DSA or STACK) that contains AUTOMATIC storage and other variables.
    - Size of the STATIC INTERNAL CSECT.
- Additional C/C++ Specific Information [z/OS V1.10 and later]
  - Significant compiler options identified
  - If the SERVICE compiler option is used to provide further user supplied version or application specific information, this information is extracted from the CSECT.
  - Does the generated code take advantage of the performance improvements provided by new hardware instruction. (i.e., What are the ARCH and TUNE compiler option values?)
  - Size of the initial STACK and required.
  - C/C++ run-time option overrides are identified and decoded for any release of C/C++.
- Additional Assembler Specific Information

---

<sup>3</sup> For the OS PL/I V2 and PL/I for MVS compilers, options are only generated for MAIN procedures, and/or for non-MAIN procedures with the TEST option specified. OS PL/I V1 and VA PL/I do not generate options. The TEST(NONE,NOSYM) is recommended, since this form of TEST does not create any execution-time overhead. The storage overhead is minimal.

## Edge Portfolio Analyzer General Information

- CSECTs that obtain the system date using the TIME system macro or the Store Clock (STCK) hardware instruction.
- CSECTs that issue LINK and/or LOAD SVCs to dynamically access separately link edited modules. In some cases, the target of the LINK/LOAD can be determined.
- CSECTs that issue ABEND macros and/or user defined SVCs.

This is only a sample of the detailed information that is crucial for many operational activities such as the successful migration to a new compiler. This information is also very useful for problem source identification and problem determination activities even when not associated with a migration project.

### ***Analyzer Derivative Reporting***

The Edge Portfolio Analyzer also provides for the creation of derivative or exception reports. This reporting technique is required when multiple CSECTs must be analyzed to reach a conclusion. Derivative reporting is used to create global cross reference reports across multiple CSECTs and across multiple load libraries in order to identify situations troublesome to LE or other migrations. These derivative reports are created using External Analyzers.

The list below identifies some of the information that can be created using derivative reporting. For more specifics on the External Analyzers provided with the Edge Portfolio Analyzer, see *External Analyzers* on Page 16.

- Global Cross Reference Information
  - List of load modules containing a specific CSECT that are found across all the libraries analyzed
  - List of languages used to access a system facility such as CICS, DB2 or IMS
  - OS/VS COBOL applications currently using CICS that will no longer run under CICS TS V3.1 or later. (These applications must be migrated to a newer COBOL compiler if they are to execute in the future.)
  - Applications written in PL/I or C/C++ and running under CICS that might be exposed to data loss or data corruption because of a problem with the use of Additional Floating Point Registers. (The AFP hardware feature provides twelve additional registers.)
  - A suite of four reports that identifies supported and unsupported COBOL or PL/I application components that will use DB2 V8 or later.
  - Identify PL/I programs that issue a FETCH to dynamically load modules and the target of the FETCH request

## Edge Portfolio Analyzer General Information

- Performance Opportunity Identification
  - Performance-critical compile-time options (i.e. OPTIMIZE, ARCH and/or TUNE)
  - COBOL programs using the TRUNC(BIN) compiler option
  - Conflicting characteristics such as COBOL programs with SORT verbs but without the FASTSRT option
  - Load modules employing run-time override and tuning options (i.e. PLIXOPT, IGZEOPT, IGZETUN, CEEUOPT)
- LE Migration Sensitivity
  - PL/I Sort
  - PL/I Fast Initialization
  - Multi-tasking
  - PLIDUMP
  - PLITEST CALLs
  - PL/I Interlanguage Call facility
  - Assembler use of LINK/LOAD SVCs
- COBOL Migration Planning
  - Mixtures of RES and NORES programs
  - Back-level library modules
  - Language level specifications
- z/OS Migration Planning
  - Identify changes or enhancements to the system and/or hardware that can be exploited or that must be updated to conform to new support requirements.

Derivative reports are important in managing and controlling a changing production environment. For examples of the types of information in the Detail Data Files, see Table 1 on Page 22.

### ***External Interfaces***

Edge Portfolio Analyzer can work in conjunction with other development tools. The detail data files are fully documented flat files that can be used as input to many host and workstation based tools commonly found in development organizations. Virtually any tool that can process and/or perform queries on character data can be used to process the detail data files. See **Figure 2**, Page 15.

## Edge Portfolio Analyzer General Information

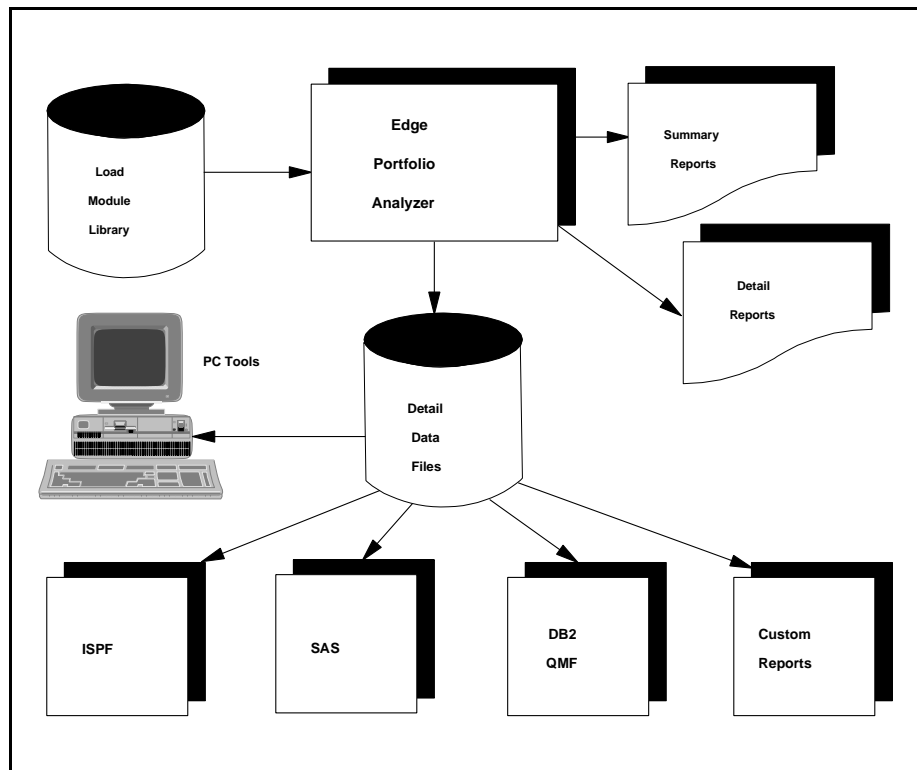


Figure 2: External Tool Interfaces

### Automated Relink

An External Analyzer provided with the Edge Portfolio Analyzer can create Linkage Editor control statements to automatically relink specific load modules or all the load modules in a given load library. With this feature, you can:

- Force replacement of IBM supplied run-time subroutines. This might be used in a COBOL application to support newly generated code in NORES or MIXRES environments or to upgrade a PL/I application to use the most recent level of the Resident Library.
- Update the run-time interfaces for subsystems such as CICS, IMS and/or DB2 to the latest level.
- Locate and propagate new versions of user written subroutines to all applications that use them. Date routines or ABEND routines might be updated this way.
- Indicate that the control statements are to be used with the LE conforming run-time libraries or with the language specific libraries.

## Edge Portfolio Analyzer General Information

Automated generation of the Linkedit control statements allows you to achieve desired improvements quickly with limited manpower requirements.

### External Tool Population

Output from the Edge Portfolio Analyzer includes machine-friendly data as well as people-friendly reports. The Analyzer's results can be used by other tools, either mainframe or PC, that you have at your disposal. Typical tools for analysis might be mainframe or PC versions of DB2/SQL or ISPF. Some examples are shown in **Figure 2**, on Page 15.

The output from the Analyzer is also commonly used to load other application development tools and/or configuration management tools. For example, a configuration management package must have the exact set of compiler options used to compile a program in order to initiate the compile process for that component. Since the Analyzer extracts that information, it can be used to populate the configuration management tables for the application.

Most of the information in the printed detail reports is also directed to the Detail Data Files. The use of these Detail Data Files avoids the need to spend unnecessary effort to parse printed reports to provide exception reporting. One record is created for each library analyzed and one record is created for each CSECT analyzed. An optional file is created with records identifying external references, load module size, alias names, CSECT-to-CSECT references, and selected types of dynamic CALLs to external routines. A file can also be generated that contains any User IDR information stored in the load module. User IDR records contain, for example, information about IBM APARs and/or PTFs that have been applied to the component.

### External Analyzers

The External Analyzers included with the Edge Portfolio Analyzer use the **Machine Friendly files** that are created when a load library is processed. These files contain essentially the same information that is in the Detail Reports printed during the primary load module analysis. (EPA is the name of this primary analysis program.) The difference is that the Machine Friendly files have the information coded to allow simple selection by a program rather than by a person reading the files.

These External Analyzer programs happen to be written in COBOL and/or PL/I, but, because the information is provided in character format, the programs could have been written in almost any programming language or report generator. Use of character only data also allows the

## Edge Portfolio Analyzer General Information

files to be downloaded or transferred to almost any platform for further analysis. The Edge provided programs use the COBOL COPY and the PL/I %INCLUDE facility to obtain the record format definitions to assure that the latest file formats are always used. Although these External Analyzers are written in COBOL and PL/I, the file formats are also provided in Assembler, DB2 and SAS for the convenience of anyone of writing custom analyzers in those languages.

In addition to the information in the primary Detail Data File (a.k.a., the **MACHINE** file because that is the DD-name for the file), there are three optional files that are created during the primary load module analysis that can be used by the External Analyzers. These optional files are the Sum File (**MRSUM**), the Cross Reference (**XREF**) File and the IDR Information File (**IDROUT**). The External Analyzers will use one or more of these Machine Friendly files depending on the objective of the program. The records in these four files are all tied together by a four-character user specified code.

The MRSUM File contains overall information about the particular analysis run. This information includes the actual load library dataset name, the volume serial number where the library resides, the date and time of the analysis, and the version and release of the EPA program that was used. The XREF file provides references from one CSECT to another within a load module, plus other miscellaneous data such as load module length, SSI values and alias information. The IDROUT file contains information from User IDR records stored in some CSECTs.

The External Analyzers to perform exception reporting are provided with the EPA in PL/I and COBOL source code programs. Sample JCL files to compile and execute the programs are also provided. One External Analyzer uses DB2 and SQL for analysis. The sample DB2 table definitions, table load and SQL examples are included. A few of the External Analyzers, because of their complexity, are only provided as object code (OCO). The External Analyzers provided as of the date of this document are described below. But, there are often new External Analyzers created between official releases of the Edge Portfolio Analyzer, and they are distributed as they become available until a new product release is distributed. If you want to determine if E/As have been updated and/or released, please check the Edge Web site or use the contact information listed in the back of each of the EPA manuals.

- **AFPREGP** - This OCO External Analyzer is used to detect PL/I and/or C/C++ applications running under CICS that may be susceptible to data loss or data corruption. (For specifics, see IBM Red Alert Flash 10417, dated 06-Oct-2996.)

- **ASMHL1x/ASMHL2x/ASMHL3x<sup>4</sup>** - This suite of three programs and three sets of JCL produces a series of outputs that can help in the impact analysis of Assembler code on an LE migration. The listings provided are:
  1. Assembler CSECTs that are found in modules that also contain High Level Language (HLL) components. The Assembler CSECTs are identified by CSECT name and load module name.
  2. An inventory of the number of observations of each Assembler CSECT across all modules and, if the input is from multiple libraries, across all libraries. This listing also indicates Assembler CSECTs that have the same name, but are not identical, because they have multiple compile dates and/or different CSECT sizes.
  3. A summary of Assembler CSECTs in descending order of number of observations of the CSECT name.
- **CBSRCHP** - This OCO E/A reports COBOL programs that contain SEARCH ALL statements. All COBOL programs that use this statement are exposed to changed behavior due to a fix to the run-time provided by installing the support for Enterprise COBOL V3.4. This support was added to LE in the base z/OS V1.7 or via APAR PQ95214. Additional APARs have been provided to minimize the impact of the change and to provide compiler messages from the Ent. COBOL V3.4.1. A complete action plan, if you have programs exposed to this change is provided in the COBOL Migration Guide (GC27-1409).
- **CICSCBx/CICSPLx** - These E/As search for OS/VS COBOL or OS PL/I V1 and V2 programs that are in load modules also containing CICS interface components. These applications are at risk if future migrations are planned to CICS TS V3.1 or later. OS/VS COBOL applications **will not run** with this new version of CICS. While the PL/I applications will still run, the IBM CICS development organization is no longer testing these components with new CICS versions. Therefore, continued use of these programs without upgrading them to newer, supported compilers should be considered as risky.
- **COBAHPx** - This program reports modules containing COBOL II or later with DATA(31) and RENT, plus at least one component having AMODE(24) that can potentially have addressing problems when migrated to OS/390 V2.9 or later from an earlier release.
- **COBBSTX** - This E/A scans for COBOL II bootstrap components that are backlevel and should be refreshed before migrating the load module to LE. The RELINKP E/A can be used to refresh the bootstrap and any other run-time components in the load module.

---

<sup>4</sup> In the following list, the "x" at the end of the program name will be either a "C" (for COBOL) or a "P" (for PL/I) depending on the source language for the program.

Edge Portfolio Analyzer  
General Information

- **COBLEZx** - This program reports modules containing only one non-LE conforming COBOL CSECT. These modules are generally more easily migrated to a newer level of COBOL.
- **COBMIGx** - This program reports CSECTs compiled by non-LE compilers that may have some migration issues based on the composition of the load module. The module contains at least one COBOL CSECT, but may contain other languages. They may also have issues related to the RES/NORES option or unexpected combinations of runtime components.
- **COBNLEx** - This program identifies COBOL applications that are not already migrated to Enterprise COBOL.
- **COBOPTx/PLIOPTx** - These two E/As provide a statistical summary report of the usage of key compiler options. These statistics can provide guidance when selecting the default options for a new compiler installation.
- **COBRNRx** - This program lists two categories of problems with COBOL and COBOL library composition.
  1. It will list load modules that contain a combination of COBOL programs compiled with both RES and NORES compiler options. This is an illegal combination when found in the same load module.
  2. Somewhat related, are load modules containing OS/VS COBOL and/or VS COBOL II programs that are all RES, but also contain runtime library CSECTs (other than bootstraps and runtime options) in the load module. The runtime library modules should not be present for RES compiled programs.
- **CSXREFx** - This program provides a cross reference listing of module names and CSECT names sorted by CSECT name.
- **DB2Q01S** - This DB2 analysis illustrates the ability to load a MACHINE and an XREF file into DB2 tables to analyze the languages that use DB2 facilities. The analysis is performed by joining the two tables and then using SQL selection criteria to match the language in the MACHINE file with the references to DB2 CSECTs in the XREF file.
- **DB2SCBx, DB2UCBx, DB2SPLx, DB2UPLx** - These four programs report COBOL and PL/I components as either supported or unsupported by DB2 Version 8 or Version 9. Any applications that are intended to exploit the new function mode (NFM) of DB2 must be at least at Enterprise COBOL V3.4 or Enterprise PL/I V3.4. This suite of analyzers is intended to identify applications that are ready for NFM with these DB2 versions or that will require some level of migration.
- **DOPTCKP** - This OCO E/A attempts to identify CEEDOPT CSECTs that were not created by IBM with the default run-time options. Modified CEEDOPTs will no longer be supported after z/OS V1.13. This E/A can help in discovering any migration or transition changes required to conform to the Statement of Direction in the z/OS V1.13 announcement.

Edge Portfolio Analyzer  
General Information

- **DUPNAMx** - This program reports instances of the same CSECT name, but with multiple compile dates and/or multiple sizes. This inconsistency check can indicate a flaw in the process used for change control, especially when there is only one source program to create the CSECT.
- **EPLIBCP** - This E/A provides a replacement for the batch compile function that was previously provided in the OS PL/I and PL/I for MVS and VM compilers, but is not available in the Enterprise PL/I Compiler. As with the old batch compile function, the individual source programs are separated by PROCESS statements. The NAME compiler option can be used to specify that the object file created by the compiler will contain a NAME record to uniquely identify the load modules that are created when the object file is used as input to the Linker/Binder. This External Analyzer is distributed in Object Code Only [OCO] format.
- **EZTPLSx** - This program identifies application components that were created in the Easytrieve Plus language.
- **GBXREFx** - This program provides a cross reference of CSECTs in load modules. The output is similar to **CSXREFx**, except it also contains the LibrID field to identify the library as well as the module name. The addition of the LibrID field allows the analysis to cover any number of libraries in one pass, whereas CSXREFx is intended for use with only a single library. The full 44-character DSName of the libraries will be listed if the MRSUM records from the EPA run are made available to the program.
- **JCLGENx** - This program uses a list of load libraries as input and creates JCL for separate Jobs to run the EPA to analyze each library. A unique **PARM='/Unnnn'** is created with a numeric value to identify each execution. EPA output file names are identified by this same unique numeric value. The default for the **nnnn** value starts at 2001 and increments one for each library in the input list. Similar JCL could be built to run a series of Jobs for any External Analyzer using the same logic. Only the program name and DD-names would have to be changed in constants in the programs.
- **PIOSCNx** - This program selects any module that uses PL/I RECORD and/or STREAM I/O statements. This can be useful in an analysis of on line program where no operating system I/O is supposed to be used.
- **PLESCNx** - This program looks for load modules that require special attention in a PL/I migration because they contain one or more of the following:
  - PL/I Interlanguage Communication (ILC)
  - PL/I FETCH or RELEASE
  - PL/I Sort usage
  - References to PLITEST
  - References to PLIDUMP
  - PL/I Shared Library components exist in the module
  - PL/I Fast Initialization components exist in the module

Edge Portfolio Analyzer  
General Information

- PL/I Multitasking usage
- **PLIFCHP** - This program detects modules that use the PL/I FETCH or RELEASE statements to dynamically load other modules. In addition, the individual PROCs that issue the FETCH and the names of the modules that are FETCHed are identified. This External Analyzer is distributed in Object Code Only [OCO] format.
- **PV1SCNx** - This program selects load modules containing compiled code from any PL/I Version 1.x compiler. These programs generally require more attention during a migration than those created by Version 2.
- **RELINKP** - This External Analyzer generates the Linkage Editor control statements to relink the load modules represented in the MACHINE file created by the Analyzer. The control statements will cause the forced replacement of language run-time library components and, optionally, DB/DC CSECTs that exist in the module. It also allows the specification of specific user CSECTs that are to be replaced during the relink process. These selected CSECTs can be replaced *in addition to* the normal relinking of the module. Or the specified CSECT can be used as the only selection criteria for relinking a module. Runtime CSECTs will be refreshed if the module is selected for relinking. The relink control statements can optionally be separated into four separate files or placed in one file. The four output files are based on the four possible combinations of linkedit options RENT and REUS. The External Analyzer options are controlled by EXEC PARMs. This External Analyzer is distributed in Object Code Only [OCO] format.
- **SYSDATP** - This External Analyzer identifies Assembler, COBOL and PL/I CSECTs that request the current system date. This program uses both the MACHINE and the XREF files. Only the PL/I source code version of this E/A is provided.
- **XMODECx** - This External Analyzer identifies load modules that do not execute in 31-bit addressing mode. i.e., A module is flagged if it contains a CSECT with an AMODE and/or RMODE that will keep the module from being run AMODE(ANY), RMODE(ANY).

**Examples of Information in Detail Data Files**

Library Level Information	Module Level Information	CSECT Level Information
Library DSName	Module Name	CSECT Name
Volume Serial No.	Load Module Size	CSECT Size
Date/Time of Analysis	Entry Point Name	Compiler Name/Version/Release
EPA Version/Release	Linkedit Date, Time	Compile Date/Time
	Linkedit Attributes: AMODE/RMODE RENT REUS Overlay Alias Names SETSSI Values System Date Requests (COBOL, PL/I) User IDR Data Middleware usage (CICS, DB2, IMS)	Compiler Options (e.g.); OPTIMIZE TEST LANGLVL (COBOL, C/C++) RES (COBOL) DYNAM (COBOL) ENDJOB (COBOL) DATA{24 31} (COBOL) NUMPROC (COBOL) TRUNC (COBOL) CMPR2 (COBOL) CMPATV1 (PL/I) REORDER (PL/I) ARCH, TUNE (PL/I, C/C++) System Date Requests (ASM, VS COBOL II or later)
		Dynamic CALL Detected (COBOL) SVC Usage (ASM) LINK, LOAD, ABEND User SVCs (200-255)

**Table 1:** Information in Detail Data Files

**NOTE:** This is not intended to be an exhaustive list, but only a selected sample of the information contained in the files.

## Chapter 3: Analyzer Reports

Two "people readable" reports are produced by the Load Module Analyzer. The first report contains details for each load module and component CSECTs selected for analysis. The second report contains summary information reflecting counts of specific attributes of the programs analyzed. The following are excerpts from sample Detail Reports and Summary Reports produced by the Edge Portfolio Analyzer.

Following the Summary Reports are some sample External Analyzer Reports from selected Edge provided External Analyzers [E/As]. They all follow the same general report format, but obviously each E/A will produce different information with different objectives.

Where possible the sample reports are either the complete reports, or they are subsets, with some lines omitted to fit in the space available. In a very few cases, the formatting, but not the content may be altered, again, to conform to the space available.

OS/VS COBOL

```

P390GQ   RW LOADMOD ANALYSIS - EDGE PORTFOLIO ANALYZER      VIR06.00 DATE: 2005.038 02/07/2005 TIME: 01.53 PAGE    1
  UID = TEST   DSNAME = EPAPROD.TESTCASE.LOAD              VOLUME = USERVL
MOD-NAME  TTRC   L-ATTR  LGTH  E.P.  L-DATE  EPNAME  ----- ATTRIBUTE FLAGS -----
CSECT-NM  LENGTH  DATE COMPILED
-----
OLDONEL1 003A112C 02E2 0004F8 000000 94.301 SMALLPGM RMODE 24 AMODE 24 ⑦
  CS-SMALLPGM 0004F4  OCT 28,1994 13.01.03 ⑥
      RES      NOSTATE NOSYMD  SYSOUT  NOTEST      ⑨ VS      ① L1
      0203
      HAS DYNAMIC REFERENCE TO ILBOSRV0
      HAS DYNAMIC REFERENCE TO ILBODSPO
      HAS DYNAMIC REFERENCE TO ILBOSRV1
OLDONEL2 003A092C 02E2 0004F8 000000 94.301 SMALLPGM RMODE 24 AMODE 24
  CS-SMALLPGM 0004F4  OCT 28,1994 11.11.22
      RES      NOSTATE NOSYMD  SYSOUT  NOTEST      VS      ② L2
      0203
      HAS DYNAMIC REFERENCE TO ILBOSRV0
      HAS DYNAMIC REFERENCE TO ILBODSPO
      HAS DYNAMIC REFERENCE TO ILBOSRV1
TC170    0028082C 02E2 0089D0 000000 90.058 TC170    RMODE 24 AMODE 24
  CS-TC170   008946  FEB 27,1990 18.18.49
      RES      NOSTATE NOSYMD  SYSOUT  NOTEST      ENDJOB  NOFLOW  OBJ370  CAPEX ③      VS      L2
      0203
  CS-ABEND   000088
  UNIVSM2   0039112C 02E2 000568 000000 94.272 UNIVSM1 RMODE 24 AMODE 24
  CS-UNIVSM1 000568  SEP 29,1994 14.52.47
      RES      NOSTATE NOSYMD  SYSOUT  NOTEST      NOENDJ  NOFLOW  OBJ370  NOOPT      VS      L2
      0203
      HAS DYNAMIC REFERENCE TO ILBOSRV0
      HAS DYNAMIC REFERENCE TO ILBODSPO
      HAS DYNAMIC REFERENCE TO ASMH1
      HAS DYNAMIC REFERENCE TO UNIVS1 ⑩
      HAS DYNAMIC REFERENCE TO ILBOSRV1
UNIVSM4   0039222C 02E2 0006E0 000000 94.272 UNIVSM1 RMODE 24 AMODE 24
  CS-UNIVSM1 000554  SEP 29,1994 15.43.11
      RES      NOSTATE NOSYMD  SYSOUT  NOTEST      NOENDJ  NOFLOW  OBJ370  NOOPT      VS      L2
      0203
      HAS DYNAMIC REFERENCE TO ILBOSRV0
      HAS DYNAMIC REFERENCE TO ILBODSPO
      HAS DYNAMIC REFERENCE TO ILBOSRV1
      HAS DYNAMIC REFERENCE TO DATANAME - CANNOT DETERMINE NAME OF SUBROUTINE
  CS-ASMH1   000188
      569623400 0101 94272  ASSEMBLER LANGUAGE CSECT
  
```

NOTES:

- ① "L1" indicates LANGLVL1 was used for this module
- ② "L2" indicates LANGLVL2 was used for this module.
- ③ CAPEX: the CAPEX/CA Optimizer package has processed this module
- ④ This is an Assembler module marked with a user description.
- ⑤ The remainder of this line contains the compiler options specified for this module.
- ⑥ This line shows the CSECT Name, length and Date/Time of compile.
- ⑦ This line shows the Load Module name, the PDS location, module length, AMODE/RMODE and the Entry Point name
- ⑧ The following is the ID, Version, and Date of the processor that created the CSECT, if it is not otherwise identified.
- ⑨ The "VS" indicates this program was compiled by the OS/VS COBOL Compiler. The four digits below are the release level.
- ⑩ These are the indications of Dynamic Call Detection.

Sample Report 1: OS/VS COBOL Detail



**VS COBOL II Release 2**

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	ATTRIBUTE	FLAGS
CSECT-NM	LENGTH	DATE	COMPILED					
AEMINCOB	0002142C	02E2	000788	000000	88.139	AEMINCOB	RMODE ANY AMODE 31	
CS-AEMINCOB	000354	05/18/88	10.31.07			NOADV APOST DATA 24 NODECK NODUMP NODYNAM NOFASRT NOFDUMP C2 1.2.0		①
						NOLIB LIST MAP NUM OBJECT NOOFFST OPT DFLT DD		
						NOPFDSN RENT RES NOSEQ SZE MAX SOURCE NOSSRNG TERM		
						NOTEST NOTRUNC DFLT RW NOVBREF XREF ZWB		
CS-IGZEBST	000430					C-II BOOTSTRAP 0201 86287 C22.0		②

**Notes:**

- ① "C2" is the code for COBOL II, the Version/Release is 1.2.0
- ② This line identifies the COBOL II Bootstrap module. The "0201 86287" fields indicate the Version of the Assembler and the date the bootstrap was assembled. The "C22.0" is an internal level code for the module.

**VS COBOL II Release 3**

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	ATTRIBUTE	FLAGS
CSECT-NM	LENGTH	DATE	COMPILED					
AESIN	0002252C	12E6	002BA0	000000	89.284	AESIN	RMODE ANY AMODE 31	
CS-AESIN	002772	10/11/89	10.05.18			NOADV APOST DATA 24 NODECK NODUMP NODYNAM NOFASRT NOFDUMP C2 1.3.0		①
						NOLIB NOLIST MAP NUM OBJECT OFFSET NOOPT DFLT DD LVL 8808		
						NMPRNP RENT RES NOSEQ SZE MAX SOURCE NOSSRNG TERM		
						TEST TRNCOPT DFLT RW NOVBREF XREF ZWB NONAME NOCMR2		
						NMCLPRI NODBCS NOAWO		
CS-IGZEBST	000428					C-II BOOTSTRAP 0201 88273 C23.0		
CS-IGZETUN	000060					② C-II STG TUNING 0201 89157		

**Notes:**

- ① This is the User Level field that can be set by the compiler installer to identify the specific level of the compiler.
- ② This is a COBOL Storage Tuning module.

**VS COBOL II Release 4**

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	ATTRIBUTE	FLAGS
CSECT-NM	LENGTH	DATE	COMPILED					
UNIVSM5	0039312C	02E2	000908	000000	94.289	UNIVSM5	RMODE ANY AMODE 31	
CS-UNIVSM5	0004B8	10/16/94	13.15.43			NOADV APOST DATA 31 NODECK NODUMP DYNAM FASTSRT NOFDUMP C2 1.4.0		
						NOLIB NOLIST MAP NONUM OBJECT OFFSET OPT DFLT DD LVL 9402		
						NMPRNP RENT RES NOSEQ SZE MAX SOURCE NOSSRNG NOTERM		
						NOTEST TRNCOPT DFLT RW NOVBREF XREF ZWB NONAME NOCMR2		
						NMCLPRI NODBCS NOAWO CALL-DN		①
CS-IGZEBST	000450					C-II BOOTSTRAP 0201 93006 C24.0		

**Note:**

- ① This program contains a CALL data-name which generates a dynamic reference to the module. The name is not available.

**Sample Report 3: VS COBOL II Detail**

## COBOL/370 Release 1.x

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	-----	ATTRIBUTE	FLAGS	-----
CSECT-NM	LENGTH	DATE	COMPILED							
COBMIN37	0035162C	02E2	0005E0	000000	94.113	MIN		RMODE ANY	AMODE 31	
CS-MIN	0005E0	04/21/94	08.19.38				ADV	QUOTE	DATA 31	NODECK NODUMP NODYNAM NOFASRT C3 010100 ①
							NOLIB	LIST	NOMAP	NONUM OBJECT NOOFFST NOOPT DFLT DD LVL
							NMPRNPF	RENT	RMODANY	SEQ SZE MAX SOURCE NOSSRNG TERM
							NOTEST	TRNCSTD	DFLT RW	NOVBREF NOXREF ZWB NONAME NOCMPR2
							NMCLPRI	NODBCS	NOAWO	NOEVENT NOCURRE
CS-IGZCBSN	0004F0	1995-06-13	11:33:00				COB/370	BSTRAP	0201	95164 LE VVRRMM: 010500
CS-CEESTART	000080						LE/370	RUN-TIME	0201	95215 ③ LE VVRRMM: 010300
CS-CEEBETBL	000020						LANGUAGE	RUNTIME	0201	95215
CS-CEESG005	000018						LANGUAGE	RUNTIME	0201	95164
CS-CEEBINT	000008						LANGUAGE	RUNTIME	0201	95215
CS-CEEBLLST	000060						LANGUAGE	RUNTIME	0201	95215
CS-CEEBTRM	0000A8						LANGUAGE	RUNTIME	0201	95215
CS-CEEBPUBT	000070						LANGUAGE	RUNTIME	0201	95215
CS-CEEARLU	0000A8						LANGUAGE	RUNTIME	0201	95215
CS-CEEBPIRA	000280						LANGUAGE	RUNTIME	0201	95215
CS-CEECPYRT	0000E8						LANGUAGE	RUNTIME	0201	95215 LE/CEE COPYRIGHT NOTICE
COB370LE	004C052C	02E2	001018	000000	96.080	TRIAL		RMODE ANY	AMODE 31	
CS-TRIAL	0004E0	03/20/96	23.31.37				ADV	APOST	DATA 31	NODECK NODUMP NODYNAM NOFASRT C3 010101 ②
							NOLIB	LIST	MAP	NONUM OBJECT NOOFFST OPTFULL DFLT DD LVL
							NMPRNPF	RENT	RMODANY	SEQ SZE MAX SOURCE NOSSRNG NOTERM
							NOTEST	TRNCSTD	DFLT RW	NOVBREF NOXREF ZWB NONAME NOCMPR2
							NMCLPRI	NODBCS	NOAWO	NOEVENT NOCURRE
CS-IGZCBSN	0004F0	1995-06-13	11:33:00				COB/370	BSTRAP	0201	95164 LE VVRRMM: 010500
CS-CEESTART	000080						LE/370	RUN-TIME	0201	95215 ③ LE VVRRMM: 010300
CS-CEEBETBL	000020						LANGUAGE	RUNTIME	0201	95215
CS-CEESG005	000018						LANGUAGE	RUNTIME	0201	95164
CS-CEEBINT	000008						LANGUAGE	RUNTIME	0201	95215
CS-CEEBLLST	000060						LANGUAGE	RUNTIME	0201	95215
CS-CEEBTRM	0000A8						LANGUAGE	RUNTIME	0201	95215
CS-CEEBPUBT	000070						LANGUAGE	RUNTIME	0201	95215
CS-CEEARLU	0000A8						LANGUAGE	RUNTIME	0201	95215
CS-CEEBPIRA	000280						LANGUAGE	RUNTIME	0201	95215
CS-CEECPYRT	0000E8						LANGUAGE	RUNTIME	0201	95215 LE/CEE COPYRIGHT NOTICE

## Note:

- ① "C3" Code is for COBOL/370 and/or COBOL for MVS and VM. "010100" is V1R1.0  
 ② "C3" Code is for COBOL/370 and/or COBOL for MVS and VM. "010101" is V1R1.1  
 ③ Even though COBOL/370 is RES-like, there are going to be CEExxxx library CSECTs in the load module.

Sample Report 4: COBOL/370 R1.x Detail

## COBOL for MVS and VM

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	ATTRIBUTE FLAGS														
CSECT-NM	LENGTH	DATE COMPILED																			
COBMT1	004E2C2C	02E2	0010A8	000000	96.232	TRIAL	RMODE	ANY	AMODE	31											
CS-TRIAL	000598	08/19/96	23.04.10	ADV	APOST	DATA	31	NODECK	NODUMP	NODYNAM	NOFASRT	C3	010200	①							
				NOLIB	LIST	MAP	NONUM	OBJECT	NOOFFST	OPTFULL	DFLT	DD	LVL								
				NMPRPF	RENT	RMODANY	SEQ	SIZE	MAX	SOURCE	NOSSRNG	NOTERM									
				NOTEST	TRNCSTD	DFLT	RW	NOVBREF	NOXREF	ZWB	NONAME	NOCMPR2									
				NMCLPRI	NODBCS	NOAWO	NOEVENT	NOCURRE						③	LE	VVRRMM:	010500				
CS-IGZCBS0	0004C8	1995-06-13	11:33:00	COB	MVS->	BSTRAP	0201	95164						LE	VVRRMM:	010300					
CS-CEESTART	000080			LE/370	RUN-TIME	0201	95215	②													
CS-CEEBETBL	000020			LANGUAGE	RUNTIME	0201	95215														
CS-CEESG005	000018			LANGUAGE	RUNTIME	0201	95164														
CS-CEEBINT	000008			LANGUAGE	RUNTIME	0201	95215														
CS-CEEBLLST	000060			LANGUAGE	RUNTIME	0201	95215														
CS-CEEBTRM	0000A8			LANGUAGE	RUNTIME	0201	95215														
CS-CEEBPUBT	000070			LANGUAGE	RUNTIME	0201	95215														
CS-CEEARLU	0000A8			LANGUAGE	RUNTIME	0201	95215														
CS-CEEBPIRA	000280			LANGUAGE	RUNTIME	0201	95215														
CS-CEECPYRT	0000E8			LANGUAGE	RUNTIME	0201	95215	LE/CEE	COPYRIGHT NOTICE												

## Note:

- ① "C3" Code is for COBOL and/or COBOL for MVS and VM. "010200" is V1R2.0
- ② Even though COBOL/370 is RES-like, there are going to be CEExxxx library CSECTS in the load module.
- ③ The version and release shown are the indicated values stored in the module by LE or the compiler that generated the CSECT.

## Sample Report 5: COBOL for MVS and VM Detail

## COBOL for OS/390 and VM

```

MOD-NAME  TTRC  L-ATTR  LGTH  E.P.  L-DATE  EPNAME  -----  ATTRIBUTE  FLAGS  -----
CSECT-NM   LENGTH  DATE COMPILED
-----
COBOMT1  006F0B2C  02E2  0010A8  000000  98.078  TRIAL    RMODE ANY AMODE 31
CS-TRIAL  000598  03/19/98  22.32.12  ADV  APOST  DATA 31 NODECK  NODUMP  NODYNAM  NOFASRT      C3 020100 ①
          NOLIB  LIST  MAP  NONUM  OBJECT  NOOFFST OPTFULL DFLT DD LVL
          NMPRNPF RENT  RMODANY SEQ  SZE MAX  SOURCE  NOSSRNG NOTERM
          NOTEST TRNCSTD DFLT RW NOVBREF NOXREF ZWB  NONAME  NOCMR2
          NMCLPRI NOCBCS NOAWO NOEVENT NOCURRE PGM(CO) INTD(A) PROGRAM ⑥
CS-IGZCBS0 0004C8 1995-06-13 11:33:00 COB MVS BSTRAP 0201 95164 ② ④ LE VRRMM: 010500
CS-CEESTART 000080 LANGUAGE RUNTIME 0201 95215 ③ ⑤
CS-CEEBETBL 000020 LANGUAGE RUNTIME 0201 95215
CS-CEESG005 000018 LE/370 RUN-TIME 0201 95164
CS-CEEBINT 000008 LANGUAGE RUNTIME 0201 95215
CS-CEEBLLST 000060 LANGUAGE RUNTIME 0201 95215
CS-CEEBTRM 0000A8 LANGUAGE RUNTIME 0201 95215
CS-CEEBPUBT 000070 LANGUAGE RUNTIME 0201 95215
CS-CEEARLU 0000A8 LANGUAGE RUNTIME 0201 95215
CS-CEEBPIRA 000280 LANGUAGE RUNTIME 0201 95215
CS-CEECPYRT 0000E8 LE/370 RUN-TIME 0201 95215 LE/CEE COPYRIGHT NOTICE

```

## Note:

- ① "C3" Code is for COBOL/370, COBOL for MVS and VM., and COBOL for OS/390 and VM "020100" is V2R1.0
- ② Note that the bootstrap module and name has changed for COBOL for OS/390 and VM from COBOL/370.
- ③ Even though COBOL for OS/390 is RES-like, there are going to be CEExxxx library CSECTs in the load module.
- ④ The two compiler options above are new for COBOL for OS/390 and VM.
- ⑤ This value is the version, release and modification level of the LE runtime module. Each LE conforming CSECT containing a full signature will contain this information.
- ⑥ The "PROGRAM" token means this is a 'program' rather than a 'class' object.

## Enterprise COBOL for z/OS and OS/390

```

P390G1  RW LOADMOD ANALYSIS - EDGE PORTFOLIO ANALYZER      2002/02A DATE: 2002.220 08/08/2002 TIME: 13.02 PAGE   66
UID = TEST  DSNAME = EPAPROD.TESTCASE.LOAD              VOLUME = USERVL
MOD-NAME  TTRC  L-ATTR  LGTH  E.P.  L-DATE  EPNAME  ----- ATTRIBUTE FLAGS -----
CSECT-NM   LENGTH  DATE COMPILED
-----
FUNCPTR1 006D102C C2E2 0097B8 000000 02.002  FUNCPTR1  RMODE ANY  AMODE 31                                RENT REUS
CS-FUNCPTR1 0049B6 01/02/02 16.57.34  ADV  QUOTE  DATA 31  NODECK  NODUMP  NODYNAM  NOFASRT  C3 030100 ①
NOLIB LIST MAP NONUM OBJECT NOOFFST NOOPT DFLT DD LVL
NMPRNPF RENT RMODANY SEQ SZE MAX SOURCE SSRANGE NOTERM
TEST TRNCSTD DFLT RW NOVBREF XREF ZWB NONAME
NMCLPRI DBCS NOAWO NOEVENT NOCURRE PGM(CO) INTD(A) PROGRAM ⑤
NODLL NOEXPOR NODATEP CALL-DN TSTNSEP ARTHCMP
NINVOKE NEXECMD NTHRD NJAVAOO NOXML
CS-IGZCBS0 000568 2001-09-28 19:15:00  COB MVS-> BSTRAP 0104 01271 ② LE VVRRMM: 030300
CS-CEESTART 0000B0 LE/370 RUN-TIME 0104 01277 LE VVRRMM: 021000
CS-CEEBETBL 000028 LANGUAGE RUNTIME 0104 01277 ④
CS-SUBPNAM3 001E28 01/02/02 16.57.27  ADV  QUOTE  DATA 31  NODECK  NODUMP  NODYNAM  NOFASRT  C3 030100
NOLIB LIST MAP NONUM OBJECT NOOFFST OPT DFLT DD LVL
NMPRNPF RENT RMODANY SEQ SZE MAX SOURCE SSRANGE NOTERM
TEST TRNCSTD DFLT RW NOVBREF XREF ZWB NONAME
NMCLPRI DBCS NOAWO NOEVENT NOCURRE PGM(CO) INTD(A) PROGRAM
NODLL NOEXPOR NODATEP TSTNSEP ARTHCMP
NINVOKE NEXECMD NTHRD NJAVAOO NOXML
CS-CEESG005 000018 LE/370 RUN-TIME 0104 01271
CS-FUNCSUBP 002398 01/02/02 16.57.31  ADV  QUOTE  DATA 31  NODECK  NODUMP  NODYNAM  NOFASRT  C3 030100
NOLIB LIST MAP NONUM OBJECT NOOFFST OPT DFLT DD LVL
NMPRNPF RENT RMODANY SEQ SZE MAX SOURCE SSRANGE NOTERM
TEST TRNCSTD DFLT RW NOVBREF XREF ZWB NONAME
NMCLPRI DBCS NOAWO NOEVENT NOCURRE PGM(CO) INTD(A) PROGRAM
NODLL NOEXPOR NODATEP TSTNSEP ARTHCMP
NINVOKE NEXECMD NTHRD NJAVAOO NOXML
CS-CEEBINT 000008 LANGUAGE RUNTIME 0104 01277 ③
CS-CEEBLLST 00005C LANGUAGE RUNTIME 0104 01277
CS-CEEBTRM 0000A4 LANGUAGE RUNTIME 0104 01277
CS-CEEBPUBT 000070 LANGUAGE RUNTIME 0104 01277
CS-CEEARLU 0000B0 LANGUAGE RUNTIME 0104 01277
CS-CEEBPIRA 0002D0 LANGUAGE RUNTIME 0104 01277
CS-CEECPYRT 0000E2 LE/370 RUN-TIME 0104 01277 LE/CEE COPYRIGHT NOTICE

```

## Note:

- ① "C3" Code is for COBOL/370, COBOL for MVS and VM., and COBOL for OS/390 and VM, Ent. COBOL for z/OS and OS/390. "030100" is V3R1.0 which indicates Enterprise COBOL. There are three Ent. COBOL routines in this sample.
- ② Note that the bootstrap module and name has changed for COBOL for OS/390 and VM and later compared to COBOL/370.
- ③ Even though COBOL for OS/390 is RES-like, there are going to be CEExxxx library CSECTs in the load module.
- ④ This value [above] is the version, release and modification level of the LE runtime module. Each LE conforming CSECT containing a full signature will contain this information.
- ⑤ The "PROGRAM" token means this is a 'program' rather than a 'class' object.

## Sample Report 7: Enterprise COBOL for z/OS and OS/390 Detail

**PL/I Version 1**

```

MOD-NAME  TTRC  L-ATTR  LGTH  E.P.  L-DATE  EPNAME  -----  ATTRIBUTE  FLAGS  -----
CSECT-NM   LENGTH  DATE COMPILED
-----
MINI4TST  00160C2C  02E2  000458  000000  93.224  PLISTART  RMODE  24  AMODE  24
CS-PLISTART  000050          PL/I OPT COMP V1 0400 93224  CGRUNTIME
CS-*SMALL11  000150  08/12/93  18.43.50  PL/I OPT COMP V1 0400 93224  CODE 0A ①
CS-*SMALL12  0000C8          PL/I OPT COMP V1 0400 93224  STATIC
CS-PLIXOPT  00005E          PL/I OPT COMP V1 0400 93224  EXTERNAL
CS-IBMBPOPT 00001C          PL/I RT OPTIONS  0400 93224  COMPILED INFO FROM PLIXOPT
          REPORT  NOSPIE  STAE  NOCOUNT  NOFLOW
          CS-IELCGOX  0000AE          PL/I OPT COMP V1 0400 93224  CGRUNTIME
          CS-IELCGOY  0000A0          PL/I OPT COMP V1 0400 93224  CGRUNTIME
          CS-SYSPINT  000020          PL/I OPT COMP V1 0400 93224  EXTERNAL
MINI5TST  0016172C  02E2  000478  000000  93.224  PLISTART  RMODE  ANY  AMODE  31
CS-PLISTART  000050          PL/I OPT COMP V1 0510 93224  CGRUNTIME
CS-*SMALL11  000154          PL/I OPT COMP V1 0510 93224  CODE 0B ②
CS-*SMALL12  0000AC          PL/I OPT COMP V1 0510 93224  STATIC
CS-PLIXOPT  00005E          PL/I OPT COMP V1 0510 93224  EXTERNAL
CS-IBMBPOPT 000040          PL/I RT OPTIONS  0510 93224  COMPILED INFO FROM PLIXOPT
          COUNT  FLOW  4  5
          ISASIZE  NONTASK=  8192  MAJOR=  8192  MINOR=  9216  TASKS=  20
          HEAP  KEEP  BELOW  SIZE  =  4096  INCREM=  5120
          CS-IELCGOG  0000B6          PL/I OPT COMP V1 0510 93224  CGRUNTIME
          CS-IELCGOH  0000A4          PL/I OPT COMP V1 0510 93224  CGRUNTIME
          CS-SYSPINT  000020          PL/I OPT COMP V1 0510 93224  EXTERNAL

```

## Notes:

- ① MINI4TST is PL/I V1R4. The "Code 0A" is an internal code for 1.4.0.
- ② MINI5TST is PL/I V1R5.1 The "Code 0B" is an internal code for 1.5.1.

Both of these modules were Linked with NCAL and therefore have no library modules included.  
For PL/I only, the "0400" and the "0510" are internal stamps on compiler generated CSECTs indicating the compiler levels as Release 4.0. and Release 5.1 respectively. This field is not used this way for other languages.

## PL/I Version 2

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	ATTRIBUTE	FLAGS
CSECT-NM	LENGTH	DATE	COMPILED					
MIN23TST	0016222C	02E2	002B08	000000	93.225	PLISTART	RMODE ANY	AMODE 31
CS-PLISTART	000080					PL/I OPT COMP V2	2300 93225	CGRUNTIME
CS-*SMALL11	0001AC		08/13/93	17:28:42		PL/I OPT COMP V2	2300 93225	CODE OF ①
						CMPATV2	NOCHGRAP	OPTIMIZ NOINTER NOFLOW NOCOUNT TEST
CS-*SMALL12	0000E0					PL/I OPT COMP V2	2300 93225	STATIC
CS-PLIXOPT	00005E					PL/I OPT COMP V2	2300 93225	EXTERNAL
CS-IBMBPOPT	0000A1					PL/I RT OPTIONS	2300 93225	COMPILED INFO FROM PLIXOPT
						NOREPORT NOSPIE	NOSTAE	
						ISASIZE	NONTASK=	819200 MAJOR= 819200 MINOR= 65536
						ISAINCR	NONTASK=	65536 MAJOR= 65536 MINOR= 65536
						HEAP	FREE	ANYWHERE SIZE = 3145728 INCREM= 1048576
CS-PLIMAIN	0001B0	②				566896201	0201 90320	ASSEMBLER LANGUAGE CSECT
						HAS POTENTIAL SVC LINK INSTRUCTION -- EXAMINE THE SOURCE NEAR LOCATION X'0000B0' TARGET IS IBMBPEPA EP=		
CS-IELCGOG	0000B6					PL/I OPT COMP V2	2300 93225	CGRUNTIME
CS-IELCGOH	0000A4					PL/I OPT COMP V2	2300 93225	CGRUNTIME
CS-SYSPINT	000020					PL/I OPT COMP V2	2300 93225	EXTERNAL
CS-IBMLLIST	000048					PL/I OPT RUNTIME	0201 90087	
CS-IBMBCH01	000218					PL/I OPT RUNTIME	0201 91059	
CS-IBMBCGT1	000088					PL/I OPT RUNTIME	0201 89269	
CS-IBMBCW01	000212					PL/I OPT RUNTIME	0201 89269	
CS-IBMBOCL1	000304					PL/I OPT RUNTIME	0201 90320	
CS-IBMBPIR1	00085D					PL/I RUNTIME2.3	0201 90087	5668-909 ⑦
CS-IBMBXITA	0000DC					PL/I OPT RUNTIME	0201 92049	
CS-IBMBLDX	000004					PL/I OPT RUNTIME	0201 90009	
CS-IBMBSE01	000138					PL/I OPT RUNTIME	0201 89269	
CS-IBMBTOC1	0000BA					PL/I OPT RUNTIME	0201 89269	
CS-IBMBTPR1	00008E					PL/I OPT RUNTIME	0201 89269	
CS-IBMBINT1	000002					PL/I OPT RUNTIME	0201 90109	
PLI23EXT	00361B2C	02E2	001340	000000	94.247	PLISTART	RMODE ANY	AMODE 31
CS-PLISTART	000080					PL/I OPT COMP V2	2300 94247	CGRUNTIME
CS-*SAMPLE1	000110		09/04/94	15:13:33		PL/I OPT COMP V2	2300 94247	CODE OF ③ MAIN ORDER EXECOPCS
						CMPATV2	NOCHGRAP	OPTIMIZ NOINTER NOFLOW NOCOUNT TEST
CS-*SAMPLE2	000080					PL/I OPT COMP V2	2300 94247	STATIC
CS-PLIMAIN	000008	④				PL/I OPT COMP V2	2300 94247	CGRUNTIME
CS-EXTSTI	000004	⑥				PL/I OPT COMP V2	2300 94247	EXTERNAL
CM-EXTST	000004	⑤				COMMON / STATIC		
CM-EXT	000004	⑤				COMMON / STATIC		
CS-IBMLLIST	000048					PL/I OPT RUNTIME	0201 90087	
CS-IBMBPFR1	000320					PL/I OPT RUNTIME	0201 90320	
CS-IBMBPIR1	00085D					PL/I RUNTIME2.3	0201 90087	5668-909 ⑦
CS-IBMBXITA	0000DC					PL/I OPT RUNTIME	0201 92049	
CS-IBMBLDX	000004					PL/I OPT RUNTIME	0201 90009	
CS-IBMBINT1	000002					PL/I OPT RUNTIME	0201 90109	

## Notes:

- ① This is a non-MAIN procedure compiled by V2R3 ("Code 0F").
- ② This PLIMAIN for a non-MAIN is the default CSECT that creates the "No MAIN Procedure..." message.
- ③ This is a MAIN procedure compiled by V2R3.
- ④ This PLIMAIN for a MAIN provides only the address of the OPTIONS(MAIN) procedure.
- ⑤ These two items are STATIC EXTERNAL variables that do not have INITIAL values. They are "Named COMMONs" rather than CSECTS.
- ⑥ This CSECT is for a STATIC EXTERNAL variable that does have an INITIAL value.
- ⑦ The IBMBPIR1 CSECT is the only module with a reliable "Release" ID.

Sample Report 9: PL/I Version 2 Detail

## PL/I Version 2 Fetch Example

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	ATTRIBUTE	FLAGS
CSECT-NM	LENGTH	DATE COMPILED						
FETCH02	0048062C	02E2	001A88	000000	95.269	PLISTART	RMODE ANY	AMODE 31
CS-PLISTART	000080					PL/I OPT	COMP V2 2300 95269	CGRUNTIME
CS-*CALLER1	0001F0		09/26/95		09:10:27	PL/I OPT	COMP V2 2300 95269	CODE OF MAIN ORDER EXECOPS
						CMPATV2	NOCHGRAP OPTIMIZ NOINTER NOFLOW	NOCOUNT TEST
CS-*CALLER2	000180					PL/I OPT	COMP V2 2300 95269	STATIC
CS-PLIMAIN	000008					PL/I OPT	COMP V2 2300 95269	CGRUNTIME
CS-IELCGMY	0000A4					PL/I OPT	COMP V2 2300 95269	CGRUNTIME
CM-STATNUL	000050					COMMON / STATIC		
CS-STATVAL	000050					PL/I OPT	COMP V2 2300 95269	EXTERNAL
CS-FILE1	00001C					PL/I OPT	COMP V2 2300 95269	EXTERNAL
CS-IBMLLIST	000048					PL/I OPT	RUNTIME 0201 90087	
CS-IBMBOCL1	000304					PL/I OPT	RUNTIME 0201 90320	
CS-*SUBTW01	0000E0		09/26/95		09:10:29	PL/I OPT	COMP V2 2300 95269	CODE OF ORDER EXECOPS
						CMPATV2	NOCHGRAP OPTIMIZ NOINTER NOFLOW	NOCOUNT TEST
CS-*SUBTW02	000078					PL/I OPT	COMP V2 2300 95269	STATIC
CS-IBMBPAF1	0000FC					PL/I OPT	RUNTIME 0201 89269	
CS-IBMBPFR1	000320					PL/I OPT	RUNTIME 0201 90320	①
CS-IBMBPIR1	00085D					PL/I R/TIME	2.3 0201 90087	5668-909
CS-IBMBXITA	0000DC					PL/I OPT	RUNTIME 0201 92049	
CS-IBMBLDX	000004					PL/I OPT	RUNTIME 0201 90009	
CS-IBMBINT1	000002					PL/I OPT	RUNTIME 0201 90109	
CS-IBMBTOC1	0000BA					PL/I OPT	RUNTIME 0201 89269	
CS-IBMBTPR1	00008E					PL/I OPT	RUNTIME 0201 89269	

## Notes:

① This is the module that indicates that a program in the load module has a FETCH and/or RELEASE statement.

FETCH03	0049022C	02E2	001538	000000	95.271	PLISTART	RMODE ANY	AMODE 31
CS-PLISTART	000080					PL/I OPT	COMP V2 2300 95271	CGRUNTIME
CS-***SUB31	000178		09/28/95		23:17:42	PL/I OPT	COMP V2 2300 95271	CODE OF ORDER EXECOPS
						CMPATV2	NOCHGRAP OPTIMIZ NOINTER NOFLOW	NOCOUNT TEST ②
CS-***SUB32	000080					PL/I OPT	COMP V2 2300 95271	STATIC
CS-PLIMAIN	0001B0					566896201 0201 90320 ASSEMBLER LANGUAGE CSECT ③		
						HAS POTENTIAL SVC LINK INSTRUCTION -- EXAMINE THE SOURCE NEAR LOCATION 'X'0000B0' TARGET IS IBMBPEPA EP=		
CS-IBMLLIST	000048					PL/I OPT	RUNTIME 0201 90087	
CS-IBMBPFR1	000320					PL/I OPT	RUNTIME 0201 90320	
CS-IBMBPIR1	00085D					PL/I R/TIME	2.3 0201 90087	5668-909
CS-IBMBXITA	0000DC					PL/I OPT	RUNTIME 0201 92049	
CS-IBMBLDX	000004					PL/I OPT	RUNTIME 0201 90009	
CS-IBMBINT1	000002					PL/I OPT	RUNTIME 0201 90109	
CS-IBMBOCL1	000304					PL/I OPT	RUNTIME 0201 90320	
CS-IBMBTOC1	0000BA					PL/I OPT	RUNTIME 0201 89269	
CS-IBMBTPR1	00008E					PL/I OPT	RUNTIME 0201 89269	

## Notes:

② This set of options is included because, while not a MAIN procedure because it is FETCHed, it does have the TEST option.

③ This PLIMAIN is included because it is not a MAIN procedure. The LINK detected is to a module that generates an error message and terminates a procedure if it were not the object of a FETCH.

Sample Report 10: PL/I Version 2 - Fetch Example Detail

PL/I for MVS and VM

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	ATTRIBUTE	FLAGS
CSECT-NM	LENGTH	DATE COMPILED						
PMXOPT1	004A152C	02E2	0017D8	000000	96.016	CEESTART	RMODE ANY	AMODE 31
CS-CEEESTART	000080					PL/I FOR MVS+VM	1110	96016
CS-CEEMAIN	000010					PL/I FOR MVS+VM	1110	96016
CS-CEEBETBL	000020					LANGUAGE RUNTIME	0201	95215
CS-CEER00TA	0001E8					LANGUAGE RUNTIME	0201	95215
CS-**TRIAL1	0000A8		01/16/96	22:48:56		PL/I FOR MVS+VM	1110	96016
						CMPATV2	NOCHGRAP	NOOPT
							NOINTER	NOFLOW
							NOCOUNT	NOTEST
CS-**TRIAL2	000098					PL/I FOR MVS+VM	1110	96016
CS-PLIXOPT	000058					PL/I FOR MVS+VM	1110	96016
CS-IBMRINP1	000028					PL/I OPT RUNTIME	0201	95205
CS-CEEUOPT	0006E0					LE/370 OPTIONS	1110	96016
							USER	OPTIONS - MACRO LEVEL = 06 @
								ABTERMENC ABEND
CS-CEE0PIPI	000268					LANGUAGE RUNTIME	0201	95215
CS-CEESG010	000068					LANGUAGE RUNTIME	0201	95205
CS-CEEP#INT	000228					LANGUAGE RUNTIME	0201	95215
CS-CEEP#CAL	000098					LANGUAGE RUNTIME	0201	95215
CS-CEEP#TRM	000168					LANGUAGE RUNTIME	0201	95215
CS-CEEARLU	0000A8					LANGUAGE RUNTIME	0201	95215
CS-CEEBINT	000008					LANGUAGE RUNTIME	0201	95215
CS-CEEBLLST	000060					LANGUAGE RUNTIME	0201	95215
CS-CEEBTRM	0000A8					LANGUAGE RUNTIME	0201	95215
CS-CEEBPUBT	000070					LANGUAGE RUNTIME	0201	95215
CS-CEEBPIRA	000280					LANGUAGE RUNTIME	0201	95215
CS-CEECPYRT	0000E8					LANGUAGE RUNTIME	0201	95215
CS-CEEPMATH	000018					LANGUAGE RUNTIME	0201	95215
								LE/CEE COPYRIGHT NOTICE
PMXOPT2	004C192C	02E2	0017D8	000000	96.089	CEESTART	RMODE ANY	AMODE 31
CS-CEEESTART	000080					PL/I FOR MVS+VM	1110	96089
CS-CEEMAIN	000010					PL/I FOR MVS+VM	1110	96089
CS-CEEBETBL	000020					LANGUAGE RUNTIME	0201	95215
CS-CEER00TA	0001E8					LANGUAGE RUNTIME	0201	95215
CS-**TRIAL1	0000A8		03/29/96	22:11:35		PL/I FOR MVS+VM	1110	96089
						CMPATV2	NOCHGRAP	NOOPT
							NOINTER	NOFLOW
							NOCOUNT	NOTEST
CS-**TRIAL2	000098					PL/I FOR MVS+VM	1110	96089
CS-PLIXOPT	000058					PL/I FOR MVS+VM	1110	96089
CS-IBMRINP1	000028					PL/I OPT RUNTIME	0201	95205
CS-CEEUOPT	0006E0					LE/370 OPTIONS	1110	96016
							USER	OPTIONS - MACRO LEVEL = 06 @
							RPTSTG	
							STACK	1000000, 100000, ANYWHERE, KEEP
								RPTOPTS
CS-CEE0PIPI	000268					LANGUAGE RUNTIME	0201	95215
CS-CEESG010	000068					LANGUAGE RUNTIME	0201	95205
CS-CEEP#INT	000228					LANGUAGE RUNTIME	0201	95215
CS-CEEP#CAL	000098					LANGUAGE RUNTIME	0201	95215
CS-CEEP#TRM	000168					LANGUAGE RUNTIME	0201	95215
CS-CEEARLU	0000A8					LANGUAGE RUNTIME	0201	95215
CS-CEEBINT	000008					LANGUAGE RUNTIME	0201	95215
CS-CEEBLLST	000060					LANGUAGE RUNTIME	0201	95215
CS-CEEBTRM	0000A8					LANGUAGE RUNTIME	0201	95215
CS-CEEBPUBT	000070					LANGUAGE RUNTIME	0201	95215
CS-CEEBPIRA	000280					LANGUAGE RUNTIME	0201	95215
CS-CEECPYRT	0000E8					LANGUAGE RUNTIME	0201	95215
CS-CEEPMATH	000018					LANGUAGE RUNTIME	0201	95215
								LE/CEE COPYRIGHT NOTICE

Notes:

- ① This is a MAIN procedure compiled by PL/I for MVS and VM ("Code 13").
- ② These runtime options are specified in PLIXOPT EXTERNAL variable above.

Sample Report 11: PL/I for MVS and VM Detail

## Enterprise PL/I for z/OS and OS/390

```

HIWRDL 006E0A2C 02E2 001280 000000 02.035 CEESTART RMODE ANY AMODE 31
CS-CEESTART 00007C ENT PL/I - Z/OS 3100 02033 CGRUNTIME LE VVRRMM: 021500
CS-CEEMAIN 00000C ENT PL/I - Z/OS 3100 02033 CGRUNTIME LE VVRRMM: 030100
CS-CEEBETBL 000024 LANGUAGE RUNTIME 0103 00088
CS-CEER00TA 0001E8 LE CONF BOOT Z/OS LE 00088 5647-A01
CS-CEE0PIPI 000263 LANGUAGE RUNTIME 0103 00088
PC-$PRIVATE 000128 2002-02-02 15:59:52 ENT PL/I - Z/OS 3100 02033 CODE: NOCSECT -> NAME N/A ① LE VVRRMM: 030100
NOSOS NOSERVICE ③
ENT PL/I - Z/OS 3100 02033 STATIC: NOCSECT -> NAME N/A LE VVRRMM: 030100
LE/370 RUN-TIME 0103 00090
LE/370 RUN-TIME 0103 00090
PL/I OPT RUNTIME 0103 00090
LE/370 RUN-TIME 0102 00293
LE/370 RUN-TIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LE/370 RUN-TIME 0103 00088 LE/CEE COPYRIGHT NOTICE
HIWRDPL 006F012C 02E2 0014E8 000000 02.035 CEESTART RMODE ANY AMODE 31
CS-CEESTART 00007C ENT PL/I - Z/OS 3100 02033 CGRUNTIME LE VVRRMM: 021500
CS-CEEMAIN 00000C ENT PL/I - Z/OS 3100 02033 CGRUNTIME LE VVRRMM: 030100
CS-CEEBETBL 000024 LANGUAGE RUNTIME 0103 00088
CS-CEER00TA 0001E8 LE CONF BOOT Z/OS LE 00088 5647-A01
CS-CEE0PIPI 000263 LANGUAGE RUNTIME 0103 00088
CS-@ST00001 000248 2002-02-02 09:56:25 ENT PL/I - Z/OS 3100 02033 CODE ② LE VVRRMM: 030100
NOSOS NOSERVICE
PL/I OPT RUNTIME 0103 00090
LE/370 RUN-TIME 0103 00088
LE/370 RUN-TIME 0102 00293
00000 NOT RECOGNIZED
00000 NOT RECOGNIZED
ENT PL/I - Z/OS 3100 02033 CGRUNTIME LE VVRRMM: 030100
ENT PL/I - Z/OS 3100 02033 CGRUNTIME LE VVRRMM: 030100
LE/370 RUN-TIME 0103 00090
569623400 0103 00088 HL ASSEMBLER LANGUAGE CSECT
LE/370 RUN-TIME 0103 00090
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LANGUAGE RUNTIME 0103 00088
LE/370 RUN-TIME 0103 00088 LE/CEE COPYRIGHT NOTICE

```

## Notes:

- ① This is a PL/I program compiled with the NOCSECT option. Thus, the CSECT-name is \$PRIVATE  
 ② This is a PL/I program compiled with EXTNAME(n), n > 7. Thus the CSECT has a compiler generated name.  
 ③ This line indicates NOSOS meaning the compiler did not build a Stored Options String.

Sample Report 12: Enterprise PL/I for z/OS and OS/390Detail



**C/370 Version 1**

```

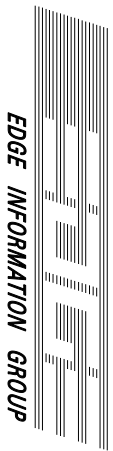
MOD-NAME  TTRC  L-ATTR  LGTH  E.P.  L-DATE  EPNAME  -----  ATTRIBUTE  FLAGS  -----
CSECT-NM   LENGTH  DATE COMPILED
-----
EDCCSMP  0049132C  C2E2  000AC8  000000  95.274  CEESTART  RMODE  24  AMODE  31  RENT REUS  ④
  CS-CEESTART  000080  C/370 V1  1101  95274  COMP GEN RUNTIME
  CS-CEEMAIN  000004  C/370 V1  1101  95274  COMP GEN RUNTIME
  CS-IBMLLIST 000048  PL/I OPT RUNTIME 0201 89076
  PC-$PRIVATE 000184  C/370 V1  1101  95274
  CS-CEESG003 000020  LE/370 RUN-TIME 0201 89131
  CS-PRINTF  00000A  566896201 0201 89129  ASSEMBLER LANGUAGE CSECT
  CS-EXIT     00000A  566896201 0201 89129  ASSEMBLER LANGUAGE CSECT
  CS-@@XINIT@ 000004  C/370 V1  1101  95274  ①
  CS-@@INIT@  000018  C/370 V1  1101  95274
  CS-@@COPYW 00003F  566896201 0201 89128  ASSEMBLER LANGUAGE CSECT
  CS-@SCINIT 00000A  566896201 0201 89129  ASSEMBLER LANGUAGE CSECT
  CS-IBMBPIR1 0003E0  PL/I RUNTIME2.21 0201 89221 5668-909 ③
  CS-IBMBINT1 000002  PL/I OPT RUNTIME 0201 89076
  CS-IBMBOCL1 0001BC  PL/I OPT RUNTIME 0201 89076 ②
  CS-IBMTOC1  0000BA  PL/I OPT RUNTIME 0201 89076
  CS-IBMBTPR1 00008E  PL/I OPT RUNTIME 0201 89076
  CS-IBMBXIT1 0000A4  PL/I OPT RUNTIME 0201 89076
  CS-TWPP005X 00001C  566896201 0201 90193  USER PLI RUNTIME EXIT

```

## Notes:

- ① This is the C/370 Compiler output
- ② Even though this is a C/370 program, it shares library routines with PL/I that are marked as PL/I.
- ③ The IBMBPIR1 CSECT is the only module with a reliable "Release" ID.
- ④ These two tokens indicate that this load module was tagged with the RENT and REUS options during Linkedit processing.

Sample Report 14: C/370 Version 1 Detail



### AD Cycle C/370 Version 1 Release 2

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	-----	ATTRIBUTE	FLAGS	-----
CSECT-NM	LENGTH	DATE COMPILED								
ERRHANDC	004A232C	02E2	0010F8	000000	95.303	FATAL@ER	RMODE	24	AMODE	31
PC-\$PRIVATE	00010C					AD CYCLE C/370		1200		95303
CS-PRINTF	00000A					566896201	0201	94037	ASSEMBLER LANGUAGE	CSECT
CS-VPRINTF	00000A					566896201	0201	94037	ASSEMBLER LANGUAGE	CSECT
CS-EDC#EXIT	00000A					C / C++ RUN-TIME	0201			94037
CS-CEESG003	00005C					LE/370 RUN-TIME	0201			94037
CS-CEESTART	0000AC					LE/370 RUN-TIME	0201			94037
CS-@@XINIT@	000004					AD CYCLE C/370		1200		95303 ①
CS-@@INIT@	000020					AD CYCLE C/370		1200		95303
CS-CEEMAIN	000030					LANGUAGE RUNTIME	0201			94037
CS-IBMLLIST	000048					PL/I OPT RUNTIME	0201			90087
CS-IBMBPIR1	000895					PL/I RUNTIME2.3	0201	94068	5668-909	③
CS-IBMBLDX	000004					PL/I OPT RUNTIME	0201			90009
CS-IBMBINT1	000002					PL/I OPT RUNTIME	0201			90109
CS-IBMBOCL1	00032C					PL/I OPT RUNTIME	0201	94068		②
CS-IBMBTOC1	0000BA					PL/I OPT RUNTIME	0201			89269
CS-IBMBTPR1	00008E					PL/I OPT RUNTIME	0201			89269
CS-IBMBXIT1	0000D8					PL/I OPT RUNTIME	0201			91268
CS-IBMBEER1	000004					PL/I OPT RUNTIME	0201			89269

LE VVRRMM: 010100

Notes:

- ① This is the AD Cycle C/370 Compiler output
- ② Even though this is a C/370 program, it shares library routines with PL/I that are marked as PL/I.
- ③ The IBMBPIR1 CSECT is the only module with a reliable "Release" ID.

Sample Report 15: AD Cycle C/370 Version 1 Release 2 Detail

### C/C++ for MVS/ESA Version 3 Release 1

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	ATTRIBUTE	FLAGS
CSECT-NM	LENGTH	DATE COMPILED						
CMVS52	006F252C	Q2E2	000A98	000000	98.164	CEESTART	RMODE ANY	AMODE 31
CS-CEESTART	000080						AD CYCLE C/370	3100 98165 COMP GEN RUNTIME
CS-CEEBETBL	000020						LANGUAGE RUNTIME	0201 95215
CS-CEER00TA	0001E8						LANGUAGE RUNTIME	0201 95215
CS-CEEMAIN	000010						AD CYCLE C/370	3100 98165 COMP GEN RUNTIME ①
PC-PRIVATE	000190						AD CYCLE C/370	3100 98165
PC-PRIVATE	000008						AD CYCLE C/370	3100 98165
CS-PRINTF	000010						569623400	0101 95215 ASSEMBLER LANGUAGE CSECT
CS-EDCINPL	000028						C / C++ RUN-TIME	0101 95215
CS-CEESG003	0000A0						LE/370 RUN-TIME	0101 95215
CS-CEEARLU	0000A8						LANGUAGE RUNTIME	0201 95215
CS-CEEBINT	000008						LANGUAGE RUNTIME	0201 95215
CS-CEEBLLST	000060						LANGUAGE RUNTIME	0201 95215
CS-CEEBTRM	0000A8						LANGUAGE RUNTIME	0201 95215
CS-CEEBPUBT	000070						LANGUAGE RUNTIME	0201 95215
CS-CEEBPIRA	000280						LANGUAGE RUNTIME	0201 95215
CS-CEECPYRT	0000E8						LE/370 RUN-TIME	0201 95215 LE/CEE COPYRIGHT NOTICE

Note:

① The token 3100 is the indicator that this is V3R1.

Sample Report 16: C/C++ for MVS/ESA Version 3 Release 1 Detail

**C/C++ for MVS/ESA Version 3 Release 2**

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	----- ATTRIBUTE FLAGS -----								
CSECT-NM	LENGTH	DATE COMPILED													
COS3911	006F332C	02E2	000A98	000000	98.169	CEESTART	RMODE	ANY	AMODE	31					
CS-CEESTART	000080						AD CYCLE	C/370	3200	98169	COMP	GEN	RUNTIME		
CS-CEEBETBL	000020						LANGUAGE	RUNTIME	0201	95215					
CS-CEER00TA	0001E8						LANGUAGE	RUNTIME	0201	95215					
CS-CEEMAIN	000010						AD CYCLE	C/370	3200	98169	COMP	GEN	RUNTIME	⓪	
PC-\$PRIVATE	000190						AD CYCLE	C/370	3200	98169					
PC-\$PRIVATE	000008						AD CYCLE	C/370	3200	98169					
CS-PRINTF	000010						569623400	0101	95215		ASSEMBLER	LANGUAGE	CSECT		
CS-EDCINPL	000028						C / C++	RUN-TIME	0101	95215					
CS-CEESG003	0000A0						LE/370	RUN-TIME	0101	95215					
CS-CEEARLU	0000A8						LANGUAGE	RUNTIME	0201	95215					
CS-CEEBINT	000008						LANGUAGE	RUNTIME	0201	95215					
CS-CEEBLLST	000060						LANGUAGE	RUNTIME	0201	95215					
CS-CEEBTRM	0000A8						LANGUAGE	RUNTIME	0201	95215					
CS-CEEBPUBT	000070						LANGUAGE	RUNTIME	0201	95215					
CS-CEEBPIRA	000280						LANGUAGE	RUNTIME	0201	95215					
CS-CEECPYRT	0000E8						LE/370	RUN-TIME	0201	95215	LE/CEE	COPYRIGHT	NOTICE		

Note:

⓪ The token 3200 is the indicator that this is V3R2.

Sample Report 17: C/C++ for MVS/ESA Version 3 Release 2 Detail

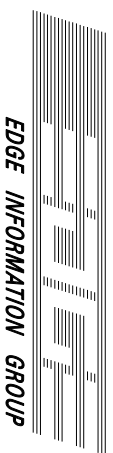
### C/C++ for OS/390 Version 1 Release 3

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	-----	ATTRIBUTE	FLAGS	-----
CSECT-NM	LENGTH	DATE COMPILED								
CPOS3913	00710F2C	C2E2	000C88	000000	98.166	CEESTART	RMODE	ANY	AMODE	31
										RENT REUS
CS-CEESTART	000080					LE/370	RUN-TIME	1300	98166	LE VVRRMM: 010300
CS-CEEBETBL	000020					LANGUAGE	RUNTIME	0201	96259	
CS-CEER00TA	0001F8					LANGUAGE	RUNTIME	0201	96261	
CS-CEEMAIN	000010					LANGUAGE	RUNTIME	1300	98166	
CS-@ST00001	0001C0					C FOR OS/390		1300	98166	Ⓣ
CS-EDCINPL	000028					C / C++	RUN-TIME	0102	96289	
CS-CEESG003	000128					LE/370	RUN-TIME	0102	96289	
CS-CBCSG003	000058					569623400	0102	96289	HL ASSEMBLER LANGUAGE	CSECT
CS-@@PPA2	000010							00000	NOT RECOGNIZED	
CS-@@XINIT@	000008					C FOR OS/390		1300	98166	
CS-@@INIT@	000090					C FOR OS/390		1300	98166	
CS-PRINTF	000010					569623400	0102	96289	HL ASSEMBLER LANGUAGE	CSECT
CS-@@GETFN	000010					569623400	0102	96289	HL ASSEMBLER LANGUAGE	CSECT
CS-CEEARLU	000088					LANGUAGE	RUNTIME	0201	96260	
CS-CEEBINT	000008					LANGUAGE	RUNTIME	0201	96260	
CS-CEEBLLST	000060					LANGUAGE	RUNTIME	0201	96259	
CS-CEEBTRM	0000B0					LANGUAGE	RUNTIME	0201	96259	
CS-CEEBPUBT	000070					LANGUAGE	RUNTIME	0201	96259	
CS-CEEBPIRA	000280					LANGUAGE	RUNTIME	0201	96261	
CS-CEECPYRT	0000F0					LE/370	RUN-TIME	0201	96261	LE/CEE COPYRIGHT NOTICE

Note:

Ⓣ The token 1300 is the indicator that this is V1R3.

Sample Report 18: C/C++ for OS/390 Version 1 Release 3 Detail



## C/C++ for z/OS Version 1 Release 13

```

P390GC  RW LOADMOD ANALYSIS - EDGE PORTFOLIO ANALYZER      VIR07.03 DATE: 2012.050 02/19/2012 TIME: 20.38 PAGE 4
  UID = C113  DSNAME = EPADEV.C4ZOS113.LOAD              VOLUME = EPADEV
MOD-NAME  TTRC  L-ATTR  LGTH  E.P.  L-DATE  EPNAME  -----  ATTRIBUTE FLAGS  -----
CSECT-NM  LENGTH  DATE COMPILED
-----
HELLO     0000C2C 02E2 000B40 000000 11.204 CEESTART RMODE ANY AMODE 31
CS-CEESTART 00007C          C/C++ Z/OS V1 0113 11199 CGRUNTIME          LE VVRRMM: 021500
CS-CEEMAIN 00000C          C/C++ Z/OS V1 0113 11199 CGRUNTIME
CS-CEEBETBL 000028          LANGUAGE RUNTIME 0106 10078
CS-CEERootA 0001F0          LE CONF BOOT Z/OS LE 10078 5694-A01
CS-STDIO   000170          C/C++ Z/OS V1 0113 11199 CODE          LE VVRRMM:Ⓞ 011300
                2011-07-18 12:29:47
                STACK:      00000160 SOS= 2 NOSERVICE
                START  NOOPT  NOCMPACT  NOHOT  PLSTHST  TARGET(LE ,Z/OS V1R13M00)
                ARCH( 5) TUNE( 5) CSECT  NORENT  NODLL  NODEBUG  DBUGHOOK(LINE)
                NOIPA  ANSIALIS UNROL(A) DBUGHOOK(BLOCK) DBUGHOOK(PATH)  DBUG(SYMBOL)
                NOGOFF NOEXPALL NOINAUTO DBUGHOOK(FUNC)  DBUGHOOK(CALL)
                ARGPA  NOCMPRS EXECOPS  CODEPAGE( 1047) NOCONVLIT
                NOIGNER NOLIB  FLOATF  FLOATNOM  FLOATNOR  FLOAT(X) NODFP  FLOAT(AFP(NVOL))
                NOUPCONV NOLONGNM ILP32  NWSIZEOF ROUND(Z)  ENUMSIZE(SMAL)
                NOROC  ROS  STRICT  NSTRCTIN NGONUMBR REDIR  PREFETCH ASSERT(RESTRICT)
                CHRUSGN BTFLUSGN AGCPNOLP NOINLINE NOHGPR  NORTCHECK
                NOXPLINK
                LANGLVL: NNEWEXCP NOLDMATH NOOLDSTR NOOLDTmplALIGN  NANSINIT
                C++ ONLY CVFT  EXH  NORTTI  OBJMODEL(COMPAT)
NAMEMANGLING(zOS12|15_DFLT)
CS-PRINTF 00000A          569623400 0106 10078 HL ASSEMBLER LANGUAGE CSECT
CS-EDCINPL 000024          C/C++ RUN-TIME 0106 10078
CS-CEESG003 00012B          LE/370 RUN-TIME 0106 10078
CS-CEEARLU 0000B8          LANGUAGE RUNTIME 0106 10078
CS-CEEBINT 000008          LANGUAGE RUNTIME 0106 10078
CS-CEEBLLST 00005C          LANGUAGE RUNTIME 0106 10078
CS-CEEBTRM 0000A4          LANGUAGE RUNTIME 0106 10078
CS-CEEBPUBT 000070          LANGUAGE RUNTIME 0106 10078
CS-CEEBPIRA 0002A0          LANGUAGE RUNTIME 0106 10078
CS-CEECPYRT 0000E2          LE/370 RUN-TIME 0106 10078 LE/CEE COPYRIGHT NOTICE

```

## Note:

Ⓞ The token 011300 is the indicator that this is V1R13. Note that, in some early V1.10 releases, the Release value was shown as X'0A' value rather than a decimal '10' value. This was fixed by IBM sometime between Release 10 and Release 11.

## Sample Report 19: C/C++ for z/OS Version 1 Release 13 Detail

**FORTRAN G/G1**

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	-----	ATTRIBUTE	FLAGS	-----
CSECT-NM	LENGTH	DATE COMPILED								
I0140	0001172C	02E2	00DDBO	000000	86.134	MAIN	RMODE	24	AMODE	24
CS-MAIN	001B98						FORTRAN G / G1	0200	86134	①
CM-	0052F0						COMMON / STATIC			②
CS-PRTAB	000CA8						FORTRAN G / G1	0200	86134	
CS-IHOFEXIT	00001C						FORTRAN MODII RT	0200	78076	
CS-IHOECOMH	000EBC						FORTRAN MODII RT	0201	79341	
CS-IHOUOPT	000338						FORTRAN MODII RT	0201	78349	
CS-IHOERRM	000624						FORTRAN MODII RT	0200	78075	
CS-IHOCOMH2	0009BD						FORTRAN MODII RT	0201	79341	
CS-FIQAP#	0006F4						5741SC103	0201	79341	FORTRAN RUNTIME - MOD II
CS-IHOFVTH	000A43						FORTRAN MODII RT	0200	78075	
CS-IHOEFNTH	000808						FORTRAN MODII RT	0200	79283	
CS-IHOEFIOS	001190						FORTRAN MODII RT	0200	79283	
CS-IHOFIOS2	000662						FORTRAN MODII RT	0200	79283	
CS-IHOUATBL	000328						FORTRAN MODII RT	0201	78349	
CS-IHOFCONI	0002FD						FORTRAN MODII RT	0200	78075	
CS-IHOFCONO	0004CA						FORTRAN MODII RT	0200	79283	
CS-IHOFTEN	000198						FORTRAN MODII RT	0200	78075	
CS-IHOETRCH	0002AE						FORTRAN MODII RT	0200	78075	

## Notes:

- ① This is the FORTRAN G/G1 Compiler output.
- ② This is a FORTRAN Blank [or unnamed] Common CSECT

Sample Report 20: FORTRAN G/G1 Detail

**FORTRAN H/H Extended**

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	-----	ATTRIBUTE	FLAGS	-----
CSECT-NM		LENGTH			DATE	COMPILED				
PS505	0002102C	02E2	0621A8	000000	92.323	MAIN		RMODE 24	AMODE 24	
CS-MAIN		05BF44						FORTRAN H / HEXT	0202 92323	①
CS-IHOECOMH		000EBC						FORTRAN MODII RT	0201 79341	
CS-IHOUOPT		000338						FORTRAN MODII RT	0201 78349	
CS-IHOERRM		000624						FORTRAN MODII RT	0200 78075	
CS-IHOCOMH2		0009BD						FORTRAN MODII RT	0201 79341	
CS-FIOAP#		0006F4						5741SC103 0201 79341	FORTRAN RUNTIME - MOD II	
CS-IHOFCVTH		000A43						FORTRAN MODII RT	0200 78075	
CS-IHOEFNTH		000808						FORTRAN MODII RT	0200 79283	
CS-IHOEFIOS		001190						FORTRAN MODII RT	0200 79283	
CS-IHOFIOS2		000662						FORTRAN MODII RT	0200 79283	
CS-IHOUATBL		000328						FORTRAN MODII RT	0201 78349	
CS-IHOFCONI		0002FD						FORTRAN MODII RT	0200 78075	
CS-IHOFCONO		0004CA						FORTRAN MODII RT	0200 79283	
CS-IHOFTEN		000198						FORTRAN MODII RT	0200 78075	
CS-IHOETRCH		0002AE						FORTRAN MODII RT	0200 78075	

## Notes:

① This is the FORTRAN H/H Extended Compiler output.

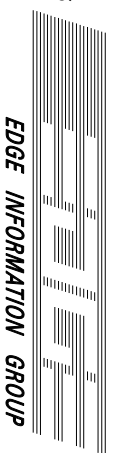
### VS FORTRAN Version 2 Release 6

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	-----	ATTRIBUTE	FLAGS	-----
CSECT-NM	LENGTH	DATE	COMPILED							
VSF2R6	006C032C	02E2	001E98	000000	98.240	MAIN#		RMODE	ANY	AMODE 31
CS-MAIN#	00043C							VS FORTRAN		0600 98240 ①
CS-AFBVIOFM	000378		10/19/93					VS FORTRAN RUN T	0206	93292
CS-AFBVCLSI	0003A0		03/25/94					VS FORTRAN RUN T	0206	94084
CS-AFBXSSQR	000224		12/06/95					VS FORTRAN RUN T	0206	95340
CS-AFBVCLMI	0001C4		10/19/93					VS FORTRAN RUN T	0206	93292
CS-AFBVINTE	000304		10/19/93					VS FORTRAN RUN T	0206	93292
CS-AFBVLCIO	000A0F							VS FORTRAN RUN T	0201	93292
CS-AFBVEMGI	000238		10/19/93					VS FORTRAN RUN T	0206	93292

Notes:

① This is the VS FORTRAN Version 2 Release 6 Compiler output.

Sample Report 22: VS FORTRAN Version 2 Release 6 Detail



## System Date Requesters

## VS COBOL II or Later Requests

```

MOD-NAME  TTRC  L-ATTR  LGTH  E.P.  L-DATE  EPNAME  -----  ATTRIBUTE FLAGS  -----
CSECT-NM   LENGTH  DATE COMPILED
-----
MT3RES    0046062C 02E2 0008A8 000000 95.233 MINTST3  RMODE  24  AMODE ANY
CS-MINTST3 0003CC 08/21/95 19.40.01 NOADV  APOST  DATA 24 NODECK  NODUMP  NODYNAM  FASTSRT  NOFDUMP C2 1.4.0
          NOLIB  LIST  MAP  NONUM  OBJECT  NOOFFST OPT  DFLT DD  LVL 9402
          NMPRNP  RENT  RES  NOSEQ  SZE MAX  SOURCE  NOSSRNG  NOTERM
          NOTEST  TRNCOPT  DFLT RW  NOVBFREF  XREF  ZWB  NONAME  NOCMR2
          NMCLPRI  NODBCS  NOAWO
          ACCEPT FROM DAY, DATE OR TIME FOUND
          566896201 0201 87238  USER ABEND MODULE
CS-ABEND   000088 566896201 0201 87238  USER ABEND MODULE
CS-IGZEBST 000450 C-II BOOTSTRAP 0201 93006 C24.0

```

## Note:

Requests from VS COBOL II, COBOL/370, COBOL for MVS and VM and/or COBOL for OS/390 and VM will be noted. Requests from other COBOL and from PL/I can be discovered using an External Analyzer provided with the Edge Portfolio Analyzer package.

## Assembler Requests

```

MOD-NAME  TTRC  L-ATTR  LGTH  E.P.  L-DATE  EPNAME  -----  ATTRIBUTE FLAGS  -----
CSECT-NM   LENGTH  DATE COMPILED
-----
CLKDEMO05 0052042C 03F2 0002B8 000000 97.192 CLKDEMO  RMODE  24  AMODE  31
CS-CLKDEMO 0002B8 569623400 0102 97192  ASSEMBLER LANGUAGE CSECT
          HAS POTENTIAL TIME LINKAGE=SYSTEM  EXAMINE THE SOURCE NEAR LOCATION X'0000BC'
          HAS POTENTIAL TIME LINKAGE=SYSTEM  EXAMINE THE SOURCE NEAR LOCATION X'000108'
          HAS POTENTIAL TIME LINKAGE=SYSTEM  EXAMINE THE SOURCE NEAR LOCATION X'000154'
          HAS POTENTIAL TIME LINKAGE=SYSTEM  EXAMINE THE SOURCE NEAR LOCATION X'0001A0'
          HAS POTENTIAL TIME LINKAGE=SYSTEM  EXAMINE THE SOURCE NEAR LOCATION X'0001EC'
          HAS POTENTIAL TIME LINKAGE=SYSTEM  EXAMINE THE SOURCE NEAR LOCATION X'000238'
          HAS POTENTIAL TIME MACRO EXPANSION - EXAMINE THE SOURCE NEAR LOCATION X'000242'
          HAS POTENTIAL TIME MACRO EXPANSION - EXAMINE THE SOURCE NEAR LOCATION X'00024C'
          HAS POTENTIAL STORE CLOCK INSTRUCTION EXAMINE THE SOURCE NEAR LOCATION X'000256'
AE110XLL 00050F2C 02E2 002220 000000 83.196 AE110  RMODE  24  AMODE  24
CS-AE110   0007B4 5734AS100 0501 83196  ASSEMBLER LANGUAGE CSECT
CS-AE110A 000388 5734AS100 0501 79351  ASSEMBLER LANGUAGE CSECT
CS-AE110B 000132 5734AS100 0501 79351  ASSEMBLER LANGUAGE CSECT
CS-AE110C 000202 5734AS100 0501 79351  ASSEMBLER LANGUAGE CSECT
CS-AE110E 00030D 5734AS100 0501 79352  ASSEMBLER LANGUAGE CSECT
CS-AE110F 00082F 5734AS100 0501 80155  ASSEMBLER LANGUAGE CSECT
          HAS POTENTIAL TIME MACRO EXPANSION - EXAMINE THE SOURCE NEAR LOCATION X'0001FC'
          HAS POTENTIAL SVC LINK INSTRUCTION -- EXAMINE THE SOURCE NEAR LOCATION X'00022C' TARGET IS CBDATE8 EP=

```

## NOTE:

These Assembler modules contain potential requests for the system date using the TIME Macro with LINKAGE=SVC (SVC 11) and LINKAGE=SYSTEM, and/or using the Store Clock (STCK) hardware instruction. There is also a sample of the LINK SVC.

Sample Report 23: System Date Requester Detail

**Alias Name Reporting**

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	ATTRIBUTE	FLAGS
CSECT-NM	LENGTH	DATE	COMPILED					
IMABEND1	00370CB1	02E2	0015F0	000000	94.069	IMABEND1	RMODE ANY AMODE 31	ALIAS OF-PLIABND1 ①
CS-IMABEND1	000FF1						566896201 0201 94069 ASSEMBLER LANGUAGE CSECT	
							HAS POTENTIAL SVC 013 INSTRUCTION EXAMINE THE SOURCE NEAR LOCATION X'0003C2'	③
							HAS POTENTIAL SVC 013 INSTRUCTION EXAMINE THE SOURCE NEAR LOCATION X'0004C6'	
CS-FPOTIME	0004D5						566896201 0201 92325 ASSEMBLER LANGUAGE CSECT	
							HAS POTENTIAL TIME MACRO EXPANSION - EXAMINE THE SOURCE NEAR LOCATION X'00004C'	
CS-DFSLI000	00011C						IMS INTERFACE 0201 93091	

\* \* \*

PLIABND1	00370C2C	02E2	0015F0	000000	94.069	IMABEND1	RMODE ANY AMODE 31	②
CS-IMABEND1	000FF1						566896201 0201 94069 ASSEMBLER LANGUAGE CSECT	
							HAS POTENTIAL SVC 013 INSTRUCTION EXAMINE THE SOURCE NEAR LOCATION X'0003C2'	③
							HAS POTENTIAL SVC 013 INSTRUCTION EXAMINE THE SOURCE NEAR LOCATION X'0004C6'	
CS-FPOTIME	0004D5						566896201 0201 92325 ASSEMBLER LANGUAGE CSECT	
							HAS POTENTIAL TIME MACRO EXPANSION - EXAMINE THE SOURCE NEAR LOCATION X'00004C'	
CS-DFSLI000	00011C						IMS INTERFACE 0201 93091	

## Notes:

- ① This is the module that is reported as an ALIAS of the module below it.
- ② This is the real module. The ALIAS is reported in the previous entry.
- ③ This CSECT indicates the presence of ABEND Macros (SVC-13)

**SETSSI Reporting**

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	ATTRIBUTE	FLAGS
CSECT-NM	LENGTH	DATE	COMPILED					
AWSSI	004F1CB3	03F2	000008	000000	97.074	IEFBR14	RMODE 24 AMODE 24	ALIAS OF-WAWSSI SSI-0123ABCD ①
CS-IEFBR14	000008						566896201 0201 89270 ASSEMBLER LANGUAGE CSECT	
WAWSSI	004F1C2E	03F2	000008	000000	97.074	IEFBR14	RMODE 24 AMODE 24	SSI-0123ABCD ②
CS-IEFBR14	000008						566896201 0201 89270 ASSEMBLER LANGUAGE CSECT	

\* \* \*

## Notes:

- ① This is the module contains a SETSSI hex value and is also reported as an ALIAS of the module below it.
- ② This is the real module with the SETSSI. Its ALIAS is reported in the previous entry.

Sample Report 24: Alias and SETSSI Reporting

## Multi-Language Application

MOD-NAME	TTRC	L-ATTR	LGTH	E.P.	L-DATE	EPNAME	ATTRIBUTE	FLAGS
CSECT-NM	LENGTH	DATE	COMPILED					
AE6992T	0006062C	12C6	018380	008DB0	89.299	AE6992T	RMODE 24	AMODE 24
CS-AE6992	007C4A	10/26/89	17.19.00			NOADV APOST	DATA 24	NODECK
						NOLIB NOLIST	MAP	NUM
						NMPPRF RENT	RES	NOSEQ
						TEST TRNCOPT	DFLT RW	NOVBREF
						NMCLPRI NOBBCS	NOAWO	
CS-AEPRT	00012A						00000	NOT RECOGNIZED
CS-IGZEBST	000428					C-II BOOTSTRAP	0201	88273
CS-AEMILE	000F10						00000	NOT RECOGNIZED
CS-MYFIX	00011C						00000	NOT RECOGNIZED
CS-AE6992T	005BAE	10/25/89	16.44.21			NOADV APOST	DATA 24	NODECK
						NOLIB NOLIST	MAP	NUM
						NMPPRF RENT	RES	NOSEQ
						TEST TRNCOPT	DFLT RW	NOVBREF
						NMCLPRI NOBBCS	NOAWO	
CS-ABEND	000088					566896201	0201	87238
CS-IHOSSCN	000208							USER ABEND MODULE
CS-IHOERRM	000624					FORTTRAN MODII	RT 0200	78075
CS-IHOFOPT	00030C					FORTTRAN MODII	RT 0200	78075
CS-IHOUOPT	000338					FORTTRAN MODII	RT 0201	78349
CS-IHOECOMH	000EBC					FORTTRAN MODII	RT 0201	79341
CS-IHOCOMH2	0009BD					FORTTRAN MODII	RT 0201	79341
CS-FIOAP#	0006F4					5741SC103	0201	79341
CS-PLISTART	000050							FORTTRAN RUNTIME - MOD II
CS-*JF69921	000FAC	06/30/89	20.30.34			PL/I OPT COMP	V1 0400	89181
CS-*JF69922	0002D8					PL/I OPT COMP	V1 0400	89181
CS-PLIMAIN	0000F4					5734AS100	0501	80162
								ASSEMBLER LANGUAGE CSECT
								HAS POTENTIAL SVC LINK INSTRUCTION -- EXAMINE THE SOURCE NEAR LOCATION X'000080'
CS-IHOSSQRT	000168					FORTTRAN MODII	RT 0200	78075
CS-IHOFCVTH	000A43					FORTTRAN MODII	RT 0200	78075
CS-IHOEFNTH	000808					FORTTRAN MODII	RT 0200	79283
CS-IHOEFIOS	001190					FORTTRAN MODII	RT 0200	79283
CS-IHOEFIOS2	000662					FORTTRAN MODII	RT 0200	79283
CS-IHOUATBL	000328					FORTTRAN MODII	RT 0201	78349
CS-IHOFCONO	0004CA					FORTTRAN MODII	RT 0200	79283
CS-IBMBCK01	000184					PL/I OPT RUNTIME	0501	80162
CS-IBMBCK01	000342					PL/I OPT RUNTIME	0501	80162
CS-IBMBIEP1	0004E0					PL/I OPT RUNTIME	0201	84352
CS-IBMBILC1	000008					PL/I OPT RUNTIME	0201	84352
CS-IBMBMAS1	0000B3					PL/I OPT RUNTIME	0501	80161
CS-IBMBMGS1	000144					PL/I OPT RUNTIME	0501	80161
CS-IBMBPIR1	0003DC					PL/I R/TIME	R40 0201	83189
CS-IHOFCONI	0002FD					FORTTRAN MODII	RT 0200	78075
CS-IHOFTEN	000198					FORTTRAN MODII	RT 0200	78075
CS-TWPP005X	00001C					5741SC103	0201	83193
CS-IBMBERR1	000750							USER PLI RUNTIME EXIT
CS-IBMBOCL1	0001AC					PL/I OPT RUNTIME	0201	84335
CS-IHOETRCH	0002AE					PL/I OPT RUNTIME	0501	80161
CS-IBMEEEF1	000002					FORTTRAN MODII	RT 0200	78075
						PL/I OPT RUNTIME	0501	80161

## Note:

- ① These are old modules that could not be identified because they predate the IDR capability. An examination of the Analyzer XREF file would reveal that they are probably old FORTRAN programs.
- ② This CSECT is the default routine if a PL/I MAIN is not present. The routine LINKs to the error routine.

## Sample Report 25: Multi-Language Application Detail

## CEEDOPT - LE V1.5 Default Run-Time Options

```

MOD-NAME  TTRC  L-ATTR  LGTH  E.P.  L-DATE  EPNAME  -----  ATTRIBUTE  FLAGS  -----
CSECT-NM   LENGTH  DATE COMPILED
-----
CEEUCT1   004C2B2C  03F2  000730  000000  96.076  CEEDOPT  RMODE ANY AMODE 31
CS-CEEDOPT 000730          LE/370  OPTIONS  0102 96076
                                     DEFAULT OPTIONS - MACRO LEVEL = 06
RESERVED
NOAIXBLD
NOALL31
BELOWHEAP 32768, 16384,FREE
NOCHECK
PLITASKCOUNT 20
ABTERMENC RETCODE
COUNTRY US
NODEBUG
ERRCOUNT 20
NOFILEHIST
ENVAR
NOFLOWC 99
HEAP 65536, 65536,ANYWHERE,KEEP
      16384, 16384
NOINQPCOPN
NOINTERRUPT
LIBSTACK 32768, 16384,FREE
MSGQ 15
MSGFILE NO DDNAME NO ENQ
      FBA
      121, 0
NATLANG ENU
ERRUNIT 6
NOOCSTATUS
NOPOSTX
NORPTSTG
NORTREUS
NOSIMVRD
STACK 524288, 524288,ANYWHERE,KEEP
STORAGE HEAP_ALLOC=X'00'
      8192
NOAUTOTASK 0
NOTRACE 4096
      X'00000000' DUMP
THREADHEAP 4096, 4096,ANYWHERE,KEEP
NOTEST ALL
THREADSTACK 524288, 524288,ANYWHERE,KEEP
NOTRAP
UPSI 00000000
NOVCTSAVE
PRTUNIT 6
XUFLOW AUTO
NOCBLOPTS
NORPTOPTS
ANYHEAP 32768, 16384,ANYWHERE,FREE
ABPERC NONE
TERMTHDACT (TRACE , N/A , )
DEPTHCONDLMT 10
NOCBLP SHPOP
NOCBLQDA
PUNUNIT 7
RDRUNIT 5
RECPAD ON
NOUSRHDLR
NOPC

```

Sample Report 26: CEEDOPT - LE V1.5 Default Run-Time Options

## CEEDOPT - LE from OS/390 V2.5 Default Run-Time Options

```

MOD-NAME  TTRC  L-ATTR  LGTH  E.P.  L-DATE  EPNAME  -----  ATTRIBUTE  FLAGS  -----
CSECT-NM   LENGTH  DATE  COMPILED
-----
CEEEOPT25 006D052C 03F2 000908 000000 98.197 CEEUOPT  RMODE ANY AMODE 31
CS-CEEUOPT 000908          LE/370  OPTIONS  0102 98197
USER      OPTIONS - MACRO LEVEL = 11
NOAIXBLD
NOALL31
BELOWHEAP      8192,      4096,FREE
CHECK
PLITASKCOUNT      20
ABTERMENC RETCODE
COUNTRY US
DEBUG
ERRCOUNT      20
FILEHIST
ENVAR
HEAP      32768,      32768,ANYWHERE,KEEP
      8192,      4096
INQPCOPN
NOINTERRUPT
LIBSTACK      8192,      4096,FREE
MSGQ      15
MSGFILE NO DDNAME NO ENQ
FBA
      121,      0
NATLANG  ENU
ERRUNIT      6
OCSTATUS
NOPOSIX
NORPTSTG
NORTEREUS
NOSIMVRD
STACK      131072, 131072,BELOW ,KEEP
STORAGE N/A, N/A, N/A
      8192
NOAUTOTASK
NOTRACE      4096
X'00000000' DUMP
THREADHEAP      4096,      4096,ANYWHERE,KEEP
NOTEST ALL
*
PROMPT
INSPPREF
TRAP
UPSI 00000000
NOVCTRSVAVE
PRTUNIT      6
XUFLOW AUTO
CBLOPTS
NONONIPTSTACK      4096,      4096,BELOW ,KEEP
NORPTOPTS
ANYHEAP      16384,      8192,ANYWHERE,FREE
ABPERC NONE
TERMTHDACT (TRACE N/A , )
DEPTHCONDLMT      10
CBLPSHPOP
CBLQDA
PUNUNIT      7
RDRUNIT      5
RECPAD ON
NOPC

```

Sample Report 27: CEEDOPT - LE from OS/390 2.5 Default Run-Time Options

**CEEUOPT - LE User Specified Options**

```
MOD-NAME  TTRC  L-ATTR  LGTH  E.P.  L-DATE  EPNAME  -----  ATTRIBUTE  FLAGS  -----
CSECT-NM   LENGTH  DATE COMPILED
-----

CEEUOPT  004C0E2C 03F2 000730 000000 96.016 CEEUOPT  RMODE ANY AMODE 31
CS-CEEUOPT 000730                LE/370 OPTIONS 0102 96016 USER  OPTIONS - MACRO LEVEL = 06
                                         NOCBLQDA

CEEUOPT1 004C242C 03F2 000730 000000 96.018 CEEUOPT  RMODE ANY AMODE 31
CS-CEEUOPT 000730                LE/370 OPTIONS 0102 96018 USER  OPTIONS - MACRO LEVEL = 06
                                         HEAP    4096,    8192, ANYWHERE, KEEP
                                         9216,    3072
                                         LIBSTACK 8192,,FREE
                                         ANYHEAP 98304,    4096, ANYWHERE, FREE
                                         NOCBLQDA
```

Sample Report 28: CEEUOPT - User Specified LE Run-Time Options

## Sample Summary Reports

### Summary Overview

### Summary Overview

```
P390GQ  RW EDGE PORTFOLIO ANALYZER - SUMMARY REPORT - VERSION V1R06.00 DATE: 2005.038 02/07/2005 TIME: 01.53 PAGE 1
      DSNAME = EPAPROD.TESTCASE.LOAD              VOLUME = USERVL
      PARM REQUESTS THAT ALL DYNAMIC REFERENCES TO COBOL ILBXXXX CSECTS BE REPORTED (/I)
      PARM REQUESTS THAT ALL STATIC LANGUAGE SUPPORT CSECTS BE ANALYZED (/A)
      PARM REQUESTS THAT MACHINE FRIENDLY REPORT FILE WILL BE GENERATED (/F) DSN=EPADDEV.V1R06M00.UTEST.MACHINE
      PARM REQUESTS THAT MACHINE FRIENDLY REPORT FILE WILL CONTAIN ID (/U) =>TEST
      IMPLICIT REQUEST FOR BASIC XREF OUTPUT DATA TO BE WRITTEN              DSN=EPADDEV.V1R06M00.UTEST.XREF
      PARM REQUESTS THAT WHO'S CALLING RECORDS BE WRITTEN TO DD-XREF (/W)
      PARM REQUESTS THAT IDR RECORDS ARE TO BE EXTRACTED AND WRITTEN TO FILE DSN=EPADDEV.V1R06M00.UTEST.IDR
      PARM REQUESTS THAT SCAN FOR TIME SVC AND STORE CLOCK INSTRUCTIONS BE PERFORMED ON UNCLASSIFIED ASSEMBLER ROUTINES (/T)
      PARM REQUESTS THAT DETAILED ANALYSIS BE PERFORMED FOR CEEUOPT AND CEEDOPT RUN-TIME OPTION SPECIFICATION MODULES (/C)
      PARM REQUESTS THAT DETAILED ANALYSIS BE PERFORMED FOR RUN-TIME CSECTS AND INFORMATION BE PLACED IN MACHINE FILE (/V)
      PARM REQUESTS SYSPRINT AND MACHINE FILE REPORTING FOR PRIVATE CODE (PC) ENTITIES (/P)
      THE FOLLOWING SPECIAL PARMS PRESENT --> 1
      NO MEMBER LEVEL SELECTION CRITERIA WERE SPECIFIED VIA THE SELECT FILE INPUT.
          TOTAL MEMBERS PROCESSED                285
          NUMBER NOT CREATED BY LINK-EDITOR        0
          NUMBER NOT EXECUTABLE                   9
          MEMBERS WOULD NOT LOAD                   0
          MODULES WITH OVERLAY STRUCTURE           0
          CSECTS NOT ANALYZED IN OVERLAY MODULES  0
          MODULES IGNORED BY DEFAULT MECHANISM    0
          MODULES WITH INTERNAL CORRUPTION         0
          MODULES WITH NOT-EDITABLE ATTRIBUTE     0
          CORRUPT PLIXOPT CSECTS - IGNORED        0
          CORRUPT IGZEOPX CSECTS - IGNORED        0
          PL/I GENED CSECTS FAILING VALIDATION     4
          MODULES IGNORED - INVALID CHARS IN NAME  0
          FAILURES IN DYN CALL PARSE OF OLD COBOL  0
```

Sample Report 29: Summary Overview

### COBOL Compiler Related Statistics

P390GQ RW EDGE PORTFOLIO ANALYZER - SUMMARY REPORT - VERSION V1R06.00 DATE: 2005.038 02/07/2005 TIME: 01.53 PAGE 2  
 DSNAME = EPAPROD.TESTCASE.LOAD VOLUME = USERVL

#### COBOL COMPILER RELATED STATISTICS

CSECTS WITH INTERNAL CONSISTENCY ERRORS	1
CSECTS CREATED BY COBOL	71
CSECTS USING ANSI V2 OR V3 COMPILER	2
CSECTS USING COBOL V4 COMPILER	0
CSECTS USING OS/VS COBOL COMPILER	27
CSECTS CREATED BY VS COBOL II	33
CSECTS CREATED BY VS COBOL II RLSE 1.0	0
CSECTS CREATED BY VS COBOL II RLSE 1.1	0
CSECTS CREATED BY VS COBOL II RLSE 2.0	6
CSECTS CREATED BY VS COBOL II RLSE 3.0	13
CSECTS CREATED BY VS COBOL II RLSE 3.E	6
CSECTS CREATED BY VS COBOL II RLSE 3.1	0
CSECTS CREATED BY VS COBOL II RLSE 3.2	0
CSECTS CREATED BY VS COBOL II RLSE 4.0	8
CSECTS CREATED BY VS COBOL II RLSE UNK	0
CSECTS CREATED BY LE CONFORMING COBOL	8
CSECTS CREATED BY COBOL/370 RLSE 1.0	1
CSECTS CREATED BY COBOL/370 RLSE 1.1	1
CSECTS CREATED BY COBOL FOR MVS V1 R2.0	1
CSECTS CREATED BY COBOL FOR MVS V1 R2.1	0
CSECTS CREATED BY COBOL FOR MVS V1 R2.2	0
CSECTS CREATED BY COBOL FOR 390 V2 R1.0	2
CSECTS CREATED BY COBOL FOR 390 V2 R1.1	0
CSECTS CREATED BY COBOL FOR 390 V2 R1.2	0
CSECTS CREATED BY COBOL FOR 390 V2 R1.3	0
CSECTS CREATED BY COBOL FOR 390 V2 R2.0	0
CSECTS CREATED BY COBOL FOR 390 V2 R2.1	0
CSECTS CREATED BY COBOL FOR 390 V2 R2.2	0
CSECTS CREATED BY IBM ENT COBOL V3 R1.0	3
CSECTS CREATED BY IBM ENT COBOL V3 R1.1	0
CSECTS CREATED BY IBM ENT COBOL V3 R2.0	0
CSECTS CREATED BY IBM ENT COBOL V3 R2.1	0
CSECTS CREATED BY IBM ENT COBOL V3 R3.0	0
CSECTS CREATED BY IBM ENT COBOL V3 R3.1	0
CSECTS CREATED BY LE COBOL: RLSE UNKNOWN	0
ODD CSECTS FROM COBOL FOR OS/390	0

Sample Report 30: COBOL Compiler Related Statistics Summary

**COBOL Compiler Specific Statistics**

P390GQ RW EDGE PORTFOLIO ANALYZER - SUMMARY REPORT - VERSION V1R06.00 DATE: 2005.038 02/07/2005 TIME: 01.53 PAGE 3  
 DSNAME = EPAPROD.TESTCASE.LOAD VOLUME = USERVL

## COBOL RUN-TIME COMPONENTS

COBOL F RUN TIME LIBRARY	0
COBOL R/T ILB CSECTS FROM OS/V5 COBOL	16
COBOL R/T ILB CSECTS FROM VS COBOL II	6
COBOL R/T ILB CSECTS FROM LE/370	6
COBOL R/T ILB CSECTS UNKNOWN ORIGIN	4
COBOL II RUNTIME TUNING (IGZETUN)	3
COBOL II RUNTIME OVERRIDES (IGZEOPT)	0
COBOL II RUNTIME DEFAULTS (IGZEOPD)	2
COBOL RUNTIME (IGZ) NOT LE CONFORMING	36
COBOL RUNTIME (IGZ) LE CONFORMING	1
COBOL II BOOTSTRAPS (IGZEBST)	29
COBOL/370-LE/370 BOOTSTRAPS (IGZCBSN/0)	5

## OS/V5, ANSI V2 AND ANSI V3 STATISTICS

OS/V5 COBOL CSECTS THAT ARE LANGLVL(1)	4
OS/V5 COBOL CSECTS THAT ARE LANGLVL(2)	23
CSECTS USING SYMDUMP OPTION	0
CSECTS USING FLOW OPTION	0
CSECTS USING STATE OPTION	2
CSECTS USING TEST OPTION	0
CSECTS USING RES OPTION	24
CSECTS USING ENDJOB OPTION	11
CSECTS USING OBJ-370 OPTION	25
CSECTS OPTIMIZED USING COBOL OPTION	5
CSECTS OPTIMIZED USING CAPEX	14
CSECTS USING RES BUT NOT ENDJOB	13
CSECTS USING COUNT OPTION - OS/V5 COBOL	0
CSECTS USING TRACE VERB - OS/V5 COBOL	0

## VS COBOL II, COBOL/370 STATISTICS

COBOL CSECTS THAT ARE CMPR2	1
COBOL CSECTS THAT ARE NOCMR2	31
CSECTS USING TEST OPTION	7
VS COBOL II CSECTS USING RES OPTION	31
CSECTS USING OPTIMIZE(STD)	31
CSECTS USING OPTIMIZE(FULL)	4
CSECTS PROCESSED BY CA-OPTIMIZER/II	0

Sample Report 31: COBOL Run-Time Components Statistics Summary

### PL/I Related Statistics

P390GQ RW EDGE PORTFOLIO ANALYZER - SUMMARY REPORT - VERSION V1R06.00 DATE: 2005.038 02/07/2005 TIME: 01.53 PAGE 4  
 DSNAME = EPAPROD.TESTCASE.LOAD VOLUME = USERVL

#### PL/I COMPILER RELATED STATISTICS

CSECTS COMPILED BY PL/I V121	0
CSECTS COMPILED BY PL/I V122	0
CSECTS COMPILED BY PL/I V123	0
CSECTS COMPILED BY PL/I V130	0
CSECTS COMPILED BY PL/I V131	0
CSECTS COMPILED BY PL/I V14	661
CSECTS COMPILED BY PL/I V150	0
CSECTS COMPILED BY PL/I V151	36
CSECTS COMPILED BY PL/I V1??	0
CSECTS COMPILED BY PL/I V2R1	0
CSECTS COMPILED BY PL/I V2R2	0
CSECTS COMPILED BY PL/I V2R2.1	0
CSECTS COMPILED BY PL/I V2R3	428
CSECTS COMPILED BY PL/I V2??	0
CSECTS COMPILED BY PL/I FOR MVS+VM R1.0	0
CSECTS COMPILED BY PL/I FOR MVS+VM R1.1	10
CSECTS COMPILED BY PL/I FOR MVS+VM R?.?	0
CSECTS COMPILED BY VA PL/I OS/390 V2.1.0	0
CSECTS COMPILED BY VA PL/I OS/390 V2.2.0	0
CSECTS COMPILED BY VA PL/I OS/390 V2.2.1	0
CSECTS COMPILED BY VA PL/I OS/390 2.2.26	0
CSECTS COMPILED BY VA PL/I OS/390 V?.?.?	0
CSECTS COMPILED BY ENT PL/I Z/OS V3.1.0	57
CSECTS COMPILED BY ENT PL/I Z/OS V3.2.0	0
CSECTS COMPILED BY ENT PL/I Z/OS V3.3.0	0
CSECTS COMPILED BY ENT PL/I Z/OS V3.4.0	0
CSECTS COMPILED BY ENT PL/I Z/OS V?.?.?	18

Sample Report 32: PL/I Compiler Related Statistics Summary

**PL/I Bootstrap and Run-Time Statistics**

P390GQ RW EDGE PORTFOLIO ANALYZER - SUMMARY REPORT - VERSION V1R06.00 DATE: 2005.038 02/07/2005 TIME: 01.53 PAGE 5  
 DSNAME = EPAPROD.TESTCASE.LOAD VOLUME = USERVL

## PL/I BOOTSTRAP AND RUN-TIME STATISTICS

PL/I V1R2	RUN TIME BOOTSTRAPS	0
PL/I V1R2.3	RUN TIME BOOTSTRAPS	0
PL/I V1R3.0	RUN TIME BOOTSTRAPS	0
PL/I V1R3.1	RUN TIME BOOTSTRAPS	0
PL/I V1R4	RUN TIME BOOTSTRAPS	46
PL/I V1R4	TASKING BOOTSTRAPS	0
PL/I V1R5.1	RUN TIME BOOTSTRAPS	1
PL/I V1R5.1	TASKING BOOTSTRAPS	2
PL/I V1R?	RUN TIME BOOTSTRAPS	0
PL/I V2R1.0	RUN TIME BOOTSTRAPS	0
PL/I V2R1.0	TASKING BOOTSTRAPS	0
PL/I V2R2.0	RUN TIME BOOTSTRAPS	0
PL/I V2R2.1	RUN TIME BOOTSTRAPS	3
PL/I V2R2.1	TASKING BOOTSTRAPS	0
PL/I V2R3	RUN TIME BOOTSTRAPS	56
PL/I V2R3	TASKING BOOTSTRAPS	0
PL/I V2R?	RUN TIME BOOTSTRAPS	0
PL/I OPT	RUNTIME (EX BOOTSTR)	1,412
PL/I(F)	- IHEXXXXX COMPONENTS	56

## Note:

PL/I Version 2 Runtime support CSECTs may also be used by C/370 programs.

Sample Report 33: PL/I Bootstrap and Run-Time Statistics Summary

## Fortran and Other Languages Statistics

P390GQ RW EDGE PORTFOLIO ANALYZER - SUMMARY REPORT - VERSION V1R06.00 DATE: 2005.038 02/07/2005 TIME: 01.53 PAGE 6  
 DSNAME = EPAPROD.TESTCASE.LOAD VOLUME = USERVL

### FORTRAN RELATED STATISTICS

FORTRAN H / H EXT GENERATED CSECTS	1
FORTRAN MOD II RUN TIME	46
FORTRAN G / G1 GENERATED CSECTS	2
FORTRAN MOD I RUN TIME	13
FORTRAN G/H (OS) RUN TIME	10
VS FORTRAN GENERATED CSECTS	1
VS FORTRAN RUN TIME	7
VS FORTRAN R/T FROM LE LIBS	0
OTHER LANGUAGE STATISTICS	
SAS/C GENERATED CSECTS	0
LATTICE C GENERATED CSECTS	0
C/370 V1 GENERATED CSECTS	15
C/370 V2 GENERATED CSECTS	0
AD CYCLE C/370 GENERATED CSECTS	34
C / C++ MVS/ESA GENERATED CSECTS	0
C / C++ OS/390 V1 GENERATED CSECTS	5
C / C++ OS/390 V2 GENERATED CSECTS	4
C / C++ Z/OS V1 GENERATED CSECTS	0
C / C++ VERSN UNKN GENERATED CSECTS	0
C / C++ RUNTIME CSECTS	9
VS PASCAL V2 GENERATED CSECTS	0
MARKV - 160 GENERATED CSECTS	0
IBM BASIC COMPILER GENERATED CSECTS	0
IBM REXX COMPILER GENERATED CSECTS	2
EZTRIEVE COMPILER GENERATED CSECTS	16

Sample Report 34: Fortran and Other Languages Statistics Summary

## Other Entity Statistics

P390GQ RW EDGE PORTFOLIO ANALYZER - SUMMARY REPORT - VERSION V1R06.00 DATE: 2005.038 02/07/2005 TIME: 01.53 PAGE 7  
DSNAME = EPAPROD.TESTCASE.LOAD VOLUME = USERVL

### OTHER ENTITY STATISTICS

UN-ATTRIBUTED LANGUAGE RUNTIME CEE...	323
LE/370 RUNTIME MODULES CEE...	160
LE CONFORMING BOOTSTRAPS: CEER00TA	27
LE ENABLED BOOTSTRAPS: CEER00TB	0
LE MULTI-TASKING BOOTSTRAPS: IBMUR00T	0
LE/370 RUNTIME DEFAULTS - CEEDOPT	6
LE/370 RUNTIME CICS/IMS REGION - CEEROPT	0
LE/370 RUNTIME OVERRIDE - CEEUOPT	9
IMS INTERFACE - DFSLI000	3
DB2 INTERFACE MODULES	0
CICS INTERFACE MODULES	0
ISPF INTERFACE MODULES	0
COBOL SAMPLES (IGZTXXXX)	0
COMPUWARE SNAP/ID MODULES	0
OPEN SOCKETS BSD4.3 INTERFACE	0
ASSEMBLER GENERATED CSECT	368
HL ASSEMBLER GENERATED CSECT	42
FORTTRAN COMMON / PL/I STATIC EXTERNAL	5
CSECTS NOT IDENTIFIED	71
USER COUNTER 1 - ABEND ROUTINES	0
USER COUNTER 2 - USER ASM SUBROUTINES	0
USER COUNTER 3 - USER EXITS TO HLL RT	0
USER COUNTER 4 - TELON RUNTIME MODULES	0
USER COUNTER 5	0
USER COUNTER 6	0
USER COUNTER 7	0
USER COUNTER 8	0

Sample Report 35: Other Entity Statistics Summary

## Sample External Analyzer Reports

### System Date Requests

#### SYSDATP: External Analyzer for System Date Requests

\*\*\* SYSDATP - YEAR 2000 - SYSTEM DATE REQUESTS - VERSION 1998/08 PAGE 1  
FOR LIBRARY: EPAPROD.TESTCASE.LOAD ON VOL=SER=USERVL, WITH /U VALUE PDOM

MODULE	CSECT	REF-TO	LIBR	COBOL	ACCEPT	PL/I	TIME	STCK	LE	SRC	
AE110XLL	AE110D		PDOM				X			MACH	①
AE110XLL	AE110F		PDOM				X			MACH	①
CLKDEMO	CLKDEMO		PDOM				X	X		MACH	①
CLKDEMO			PDOM				X	X		MODL	①
CSXREFC3	CSXREFC		PDOM	C-N	X					MACH	⑤
CSXREFC3			PDOM	N	X					MODL	③
EPN0114	IBMBJDT1	**80161*	PDOM	POTENTIAL	BAD	PL/I	DATE	BIF		MACH	④
EPN0114	*****P2	PLIDATE	PDOM				X			XREF	②
EPN0114	*****P2	PLITIME	PDOM				X			XREF	②
EPN0114			PDOM				O			MODL	
EPN0123	*****P2	PLIDATE	PDOM				X			XREF	
EPN0123	*****P2	PLITIME	PDOM				X			XREF	
EPN0123			PDOM				N			MODL	
FBM115	MRLTMDT3	CEELOCT	PDOM						X	XREF	
FBM115	MRLTMDT3	CEEGMTO	PDOM						X	XREF	
FBM115	MRLXDATC	CEELOCT	PDOM						X	XREF	
FBM115	MRLXDATC	CEEGMTO	PDOM						X	XREF	
FBM115	MRLXDATC	CEEDAYS	PDOM						X	XREF	
FBM115			PDOM	N					X	MODL	
FPM2180U	FPOGGMX		PDOM				X			MACH	
FPM2180U			PDOM				N	X		MODL	
GCM070	XDATE		PDOM				X			MACH	
GCM070	XTIMEDAT	COBDATSR	PDOM	0						XREF	
GCM070	XTIMEDAT	COBTIMSR	PDOM	0						XREF	
GCM070			PDOM	0			X			MODL	③
GCM140	GCM140	COBACDAY	PDOM	0						XREF	
GCM140	GCM140	COBACTIM	PDOM	0						XREF	
GCM140	XTIMEDAT	COBDATSR	PDOM	0						XREF	
GCM140	XTIMEDAT	COBTIMSR	PDOM	0						XREF	
GCM140			PDOM	0						MODL	
GCM240	GCM240		PDOM	C-N	X					MACH	⑤
GCM240	XTIMEDAT	COBDATSR	PDOM	0						XREF	
GCM240	XTIMEDAT	COBTIMSR	PDOM	0						XREF	
GCM240			PDOM	NO	X					MODL	
TC170	TC170	COBACDAY	PDOM	DO						XREF	⑥
TC170			PDOM	0						MODL	
TCWCA	MINTST3		PDOM	C-N	X					MACH	⑤
TCWCA			PDOM	N	X					MODL	③

Notes:

- ① These Assembler CSECTs contain Store Clock and TIME Macro uses
- ② These CSECTs contain references to PL/I Builtin Functions [BIFs] that return date and/or time values.
- ③ MODL lines are summaries of modules that were determined to contain date references. The type of reference and/or language is indicated by codes in the appropriate columns.
- ④ This CSECT is the runtime routine for the PL/I DATE BIF. In this case, the level of routine is suspected to be in error and requires an update to return correct value in the Year 2000 and beyond.
- ⑤ These COBOL II or later programs obtain an ACCEPT statement for the DATE, DAY or TIME.
- ⑥ This OS/VS COBOL program contains a dynamic reference, due to the RES compiler option, to the ACCEPT FROM DAY runtime routine.

### Sample Report 36: System Date Requests External Analyzer

Edge Portfolio Analyzer  
General Information

System Date Request Summary

SYSDATP: External Analyzer System Date Request Summary

\*\*\* SYSDATP - YEAR 2000 - SYSTEM DATE REQUESTS - VERSION 1998/08 PAGE 2  
FOR LIBRARY: EPAPROD.TESTCASE.LOAD ON VOL=SER=USERVL, WITH /U VALUE PDOM

TOTAL MODULES ANALYZED =	7366	MODULES
TOTAL COBOL CSECTS =	8746	CSECTS
TOTAL PL/I CSECTS =	165	CSECTS
TOTAL ASSEMBLER CSECTS =	12371	CSECTS
STATISTICS CELL 0	TOTAL MODULES CONTAINING COBOL	5884
STATISTICS CELL 2	MODULES CONTAINING OLD COBOL	2147
STATISTICS CELL 3	MODULES CONTAINING NEW COBOL	4757
STATISTICS CELL 4	MODULES WITH COBOL DATE REQUESTS	2058 ①
STATISTICS CELL 5	OLD COBOL CSECTS (OLD + VERY OLD)	3069
STATISTICS CELL 6	NEW COBOL CSECTS (COB II + COB/370)	5677
STATISTICS CELL 8	CSECTS W/COB II OR COB/370 ACCEPTS	709 ②
STATISTICS CELL 10	TOTAL MODULES CONTAINING PL/I	165
STATISTICS CELL 11	MODULES CONTAINING PL/I (F)	10
STATISTICS CELL 12	MODULES CONTAINING PL/I V1	161
STATISTICS CELL 13	MODULES CONTAINING PL/I V2, MVS/VM	4
STATISTICS CELL 14	MODULES W/ PL/I DATE REQUESTS	8 ③
STATISTICS CELL 15	PL/I V1 CSECTS	161
STATISTICS CELL 16	PL/I V2, PL/I MVS & VM CSECTS	4
STATISTICS CELL 19	PL/I (F) CSECTS	79
STATISTICS CELL 20	TOTAL MODULES CONTAINING ASSEMBLER	4848
STATISTICS CELL 21	MODULES CONTAINING ASM DATE REQUESTS	775 ④
STATISTICS CELL 22	ASSEMBLER CSECTS	12371
STATISTICS CELL 23	ASM CSECTS W/ TIME SVC, STORE CLOCK	779 ⑤
STATISTICS CELL 30	TOTAL MODULES CONTAINING LE LANGUAGE	7164
STATISTICS CELL 31	MODULES CONTAINING LE DATE SERVICES	544 ⑥
STATISTICS CELL 32	CSECTS USING LE DATE SERVICES	1680 ⑦
STATISTICS CELL 39	POTENTIAL BAD PL/I DATE FUNCTIONS	8 ⑧

Notes:

- ① This is a count of the total number of modules that contain COBOL date requests.
- ② This is a count of the total number of COBOL II or later programs that contain date requests.
- ③ This is a count of the total number of modules that contain PL/I date requests.
- ④ This is a count of the total number of modules that contain Assembler date requests.
- ⑤ This is a count of the total number of Assembler programs that contain date requests.
- ⑥ This is a count of the total number of modules that contain Language Environment date requests.
- ⑦ This is a count of the total number of programs that contain calls to LE date services.
- ⑧ This is a count of the total number of modules that require relinking to repair the bad PL/I DATE BIF.

Sample Report 37: System Date Request Summary



Edge Portfolio Analyzer  
General Information

Migration Assist External Analyzers

**CSECT Consistency Report**

EDGE PORTFOLIO ANALYZER: DUPNAMP 2000/07A - 2000/09/24 13:49 PAGE 1  
\*\*\*\* CSECT CONSISTENCY REPORT \*\*\*\*

CSECT	COMPILE DATES	SIZE VARIATIONS
BADOUT	2	2
BADRECS	3	2
CAUSIN	2	2
CERTS	2	2
CONVIN	2	2
CVDATE	3	2

EDGE PORTFOLIO ANALYZER: DUPNAMP 2000/07A - 2000/09/24 13:49 PAGE 2  
\*\*\*\* CSECT CONSISTENCY REPORT \*\*\*\*

TOTAL CSECTS ANALYZED 66  
UNIQUE CSECT NAMES IDENTIFIED 24  
UNIQUE CSECT NAMES WITH DIFFERENT COMPILE DATES OR SIZES 6

Notes:  
This report indicates inconsistencies that may exist because programs with the same name have different sizes and/or were compiled on different dates.

**COBOL RES / NORES Conflict Report**

EDGE PORTFOLIO ANALYZER: COBRNRP 2000/07A - 2000/09/26 00:13 PAGE 1  
\*\*\*\* COBOL WITH INVALID COMBINATIONS OF RES/NORES \*\*\*\*

LOAD MODULE	COBOL CSECTS	COBOL RUNTIME	RES OPTION	COMMENTS
AIR104	MULTIPLE COBOL	Y	RES/NORES	REQUIRES INVESTIGATION ①
AJW020	MULTIPLE COBOL	N	RES/NORES	REQUIRES INVESTIGATION
ARS020	MULTIPLE COBOL	Y	RES/NORES	REQUIRES INVESTIGATION
ARS030	SINGLE:ARS03A	Y	RES	REQUIRES INVESTIGATION
AUD300	MULTIPLE COBOL	Y	RES	REQUIRES INVESTIGATION ②

EDGE PORTFOLIO ANALYZER: COBRNRP 2000/07A - 2000/09/26 00:13 PAGE 2  
\*\*\*\* COBOL WITH INVALID COMBINATIONS OF RES/NORES \*\*\*\*

TOTAL MODULES ANALYZED 49  
MODULES CONTAINING INVALID RES/NORES COMBINATION 3  
MODULES HAS COBOL RUNTIME LINKED IN BUT ARE RES ONLY 2

NOTE: COBOL RUNTIME CSECTS ARE NOT CONSIDERED WHEN CLASSIFYING  
A MODULE AS HAVING A SINGLE COBOL CSECT

Notes:  
① This indicates a module that contains an illegal combination of the RES and the NORES Compiler option.  
② This module contains only RES components, but also has unneeded library runtime routines.

Sample Report 38: CSECT Consistency Report, COBOL RES/NORES Conflict Report

Edge Portfolio Analyzer  
General Information

PLESCNx: PL/I LE Migration Report

<b>PLESCNx: PL/I LE Migration Report</b>									
EDGE PORTFOLIO ANALYZER: PLESCNC V1R05.08 - 2004/05/17 19:27 PAGE 1									
**** FIND PL/I MIGRATION ISSUES ****									
LOAD MODULE	ILC	FETCH RELEASE	PLI- SORT	PLI- TEST	PLI- DUMP	FAST INIT	SHR LIB	MULTI TASK	
ACM007			X						
ACM030					X				
ACM070		X							
ACM077			X						
ACM107			X						
ACM130		X			X				
ACM707			X						
ACM777			X						
ACM948				X					
ACM950				X					
ACM953	X								
AWP140	X					X		X	
AWP150	X						X		
AWP160	X								
AWP170							X		
AWP180						X			
AWP190	X								
EDGE PORTFOLIO ANALYZER: PLESCNC V1R05.08 - 2004/05/17 19:27 PAGE 2									
**** FIND PL/I MIGRATION ISSUES ****									
TOTAL MODULES ANALYZED									33
MODULES CONTAINING PL/I INTERLANGUAGE COMMUNICATIONS									5
MODULES CONTAINING FETCH/RELEASE STATEMENTS									2
MODULES CONTAINING PLISORT									5
MODULES CONTAINING PLITEST									2
MODULES CONTAINING PLIDUMP									2
MODULES CONTAINING FAST INITIALIZATION									2
MODULES CONTAINING SHARED LIBRARY									2
MODULES CONTAINING MULTITASKING									1

Sample Report 39: PLESCNx: PL/I LE Migration Report



Edge Portfolio Analyzer  
General Information

CICSCBx: CICS V3.1 (OR LATER) MIGRATION ISSUES: OS/V3 COBOL

CICSCBx: CICS V3.1 (OR LATER) MIGRATION ISSUES: OS/V3 COBOL						
EDGE PORTFOLIO ANALYZER: CICSCBX V1R06.00 - YY05/01/04 16:15						PAGE 1
**** FIND CICS V3.1 (OR LATER) MIGRATION ISSUES: OS/V3 COBOL						****
LOAD MODULE	CONTAINS CSECT	MR TYPE	LINK OR COMPILE DATE	COMPILE DATE	COMPILE TIME	LIBR ID
LOADMOD1			03002			2001
	CSECT832	VS	99002	01/02/99	03:04:07	
	PRTRTN	VS	93364	12/30/93	10:23:45	
	CALCPFB3	VS	99021	01/21/99	12:34:45	
	PRESORT1	VS	95032	02/01/95	21:32:32	
LOADMOD2			03002			2001
	PASTHIST	VS	99012	01/12/99	03:14:17	
LOADMOD3			03002			2001
	LEFTOVER	VS	99002	01/02/99	03:04:27	
LOADMOD4			03002			2001
	RECALCAB	VS	98033	02/02/99	03:04:07	
	RECALCFG	VS	98033	02/02/99	03:05:34	
	RECALCJK	VS	98033	02/02/99	03:04:03	
	RECALCLM	VS	98033	02/02/99	03:02:07	
	RECALCPQ	VS	98033	02/02/99	03:04:07	
EDGE PORTFOLIO ANALYZER: CICSPLX V1R06.00 - YYYY/MM/DD HH:MM						PAGE 2
**** FIND CICS V3.1 (OR LATER) MIGRATION ISSUES: OS/V3 COBOL						****
TOTAL MODULES ANALYZED						8,476
TOTAL CSECTS ANALYZED						12,954
MODULES WITH CICS AND OS/V3 COBOL						4
CSECTS CREATED BY OS/V3 COBOL IN MODULES WITH CICS						11
NOTE: OS/V3 COBOL DOES NOT RECORD THE COMPILE DATE MM/DD/YY UNDER CERTAIN CIRCUMSTANCES						

Sample Report 40: CICSCBx: CICS V3.1 Migration Issues: OS/V3 COBOL



Edge Portfolio Analyzer  
General Information



## Chapter 4: Operating Environment

The Edge Portfolio Analyzer (EPA) will execute under OS/390 or z/OS. There are no hardware requirements other than that required by the operating system level needed to create the code in the load libraries being analyzed.

Some new instructions from the 'Extended-Immediate' feature of the z/Architecture [ARCH(7)] have been used to analyze z/OS and language attributes that are only present in z/OS V1.10 and later. The z/9 systems have these instructions. Unless you have components from these systems, you should not see any impact from these instructions.

For example, if the library contains C/C++ programs created by z/OS V1.10, the code that performs the analysis of these components may use instructions that were introduced in the extended-immediate facility.

Starting with EPA V1R05.00 (a.k.a. V2002/02A), the installation uses the *IBM Product Registration* process and will require an active entry in an **IFAPRDxx** member of PARMLIB. Execution of the EPA program will check these registration parameters. The use of the Analyzer requires no other system modifications or special authorizations other than the ability to read and load the load modules in the designated libraries.

The **load modules** to be analyzed must reside in a standard Partition Data Set (PDS). To analyze **program objects** that reside in the Partition Data Set Extended (PDSE) format supported by MVS/ESA and MVS/DFP or DFSMS, use the MVS IEBCOPY utility to copy the library from the PDSE to a PDS. There are some cases where the component in the PDSE was not created as a *load module*, but rather as a *program object*. Some program objects use facilities of the Binder that are not supported for load modules. Unfortunately, these components cannot be transferred directly to a PDS for EPA analysis. It is also possible that, because of damaged or corrupted load modules, invalid load modules, or because of unforeseen or unanticipated data, every load module may not be able to be processed successfully. For these situations, a bypass mechanism and a restart option is provided to allow the remainder of the library to be processed.

The user can provide a table that reflects the attributes of particular (by name) Assembler language subroutines. If provided, the descriptions in the table will be used to supplement the reported attributes of the identified modules in the detail report, to classify the identified modules in the Summary Report and to identify user requests to specified SVCs. An Assembler that supports the AMODE and RMODE pseudo operations is required to customize this table. Assembler H Version 2 or the new

Edge Portfolio Analyzer  
General Information

High Level Assembler from IBM are examples of translators that will meet this requirement.

Record definitions for the Detail Data Files are provided in source format for Assembler, COBOL, DB2, PL/I and SAS. These definitions should be copied into user external analyzer programs, as they will contain the current format of the Detail Data Records, Cross Reference Records, IDR User Data Records and Summary Records. The contents of each of these records is described in the Program Description and Operations Manual that is provided with the product.

Most of the sample External Analyzer (E/A) programs that perform derivative reporting are distributed in source format. The E/As provided in source format must be compiled at the customer site. These programs are written in COBOL and/or PL/I. The COBOL programs require VS COBOL II Release 3.0 or later with the NOCMR2 compiler option. The PL/I programs require the OS PL/I Optimizing Compiler Version 2 Release 3 or later. These programs can also be processed by workstation COBOL and/or PL/I compilers that are compatible with the above mainframe compilers. If the data is downloaded from the mainframe system to a workstation, the analysis can be done offline. One external analyzer tool uses DB2 and requires loading the Detail Data File and the XREF file into DB2 tables and then selecting the appropriate records using SQL.

Several External Analyzers are provided in object code only (OCO) format. The logic of these programs is extremely complex and very susceptible to errors when modified. These programs are written in PL/I and compiled with either the PL/I for MVS and VM compiler or the Enterprise PL/I Compiler. They can be linked with any current level of LE. These OCO External Analyzers include the ReLink Statement Generator (RELINKP) and the PL/I Fetch Analysis Program (PLIFCHP) programs. A third OCO E/A is a utility that can be used to batch process Enterprise PL/I compiles (EPLIBCP). This batch compile function is not provided by the Enterprise PL/I Compiler. Therefore we provide this utility as an OCO E/A to ease the migration to the new compiler for those PL/I shops that depend on this function. A fourth program (AFPREGP) is used to detect PL/I and/or C/C++ applications running under CICS that may be susceptible to data loss or data corruption. (For specifics, see IBM Red Alert Flash 10417, dated 06-Oct-2006.) Other OCO E/As may be added as part of interim EPA release updates. These will be documented in the README.1ST file that is provided for each EPA release.

This release of the EPA, V1.7.3, was tested under z/OS V1.13. The E/As have been successfully compiled with Enterprise COBOL V4.2 and Enterprise PL/I V4.2.

## Appendix A: User Feedback

We are interested in your comments about the use of the **Edge Portfolio Analyzer™** and the documentation. If you have comments, suggestions and/or requirements, please let us know. If you have questions or problems, don't suffer in silence; let us help you. Feel free to call or **E-mail [preferred]** us at one of the address listed below:

M. Carl Gehr, Jr.  
Edge Information Group  
8556 Wyoming Club Drive  
Cincinnati, OH 45215-4243

Phone:  
(513) 948-8906

Fax:  
(513) 948-8991

**E-Mail Address:**  
**[cgehr@edge-information.com](mailto:cgehr@edge-information.com)**

# **Edge Portfolio Analyzer**

## **General Information Manual**

© Copyright 1994 - 2012 Edge Information Group. All rights reserved.

This document printed from: GIM-173X.WPD on March 4, 2012

