IBM Library Server Print Preview

DOCNUM = GC26-8052-00
DATETIME = 02/07/95 11:10:48
BLDVERS = 1.2
TITLE = PL/I VSE Fact Sheet
AUTHOR =

PATH = /home/webapps/epubs/htdocs/book

FRONT

VSE/ESA mainframe PL/I joins Language Environment

PL/I for VSE/ESA

___ Executive Summary _____

IBM (*) PL/I for VSE/ESA (*) (PL/I VSE) enables you to integrate your PL/I applications into IBM Language Environment (*) for VSE/ESA (LE/VSE).

In addition to PL/I's already impressive features, you gain access to LE/VSE's rich set of library routines and enhanced interlanguage communication (ILC) with IBM COBOL for VSE/ESA (COBOL/VSE).

PL/I VSE (program number 5686-069) brings to VSE the functions of IBM PL/I MVS & VM (PL/I MVS & VM) while retaining close source compatibility with the DOS PL/I Optimizing Compiler (DOS PL/I).

PL/I is a programming language for scientific, engineering, commercial, and systems programming tasks. PL/I includes the best features of FORTRAN, COBOL, and ALGOL, with powerful structured programming constructs and many more features that other languages have only recently implemented or considered.

(*) CICS/VSE, IBM, Language Environment, MVS/ESA, Operating System/2, OS/2, Operating System/400, OS/400, SQL/DS, VM/ESA, and VSE/ESA are trademarks of the International Business Machines Corporation.

1.0 Meeting your application development needs

Today your enterprise needs efficient, consistent, and simple ways to develop quality applications and to maintain its existing storehouse of applications. IBM's goal is to provide ways for you to modularize and reuse code, as well as to take advantage of new development tools as they become available.

PL/I cross-platform portability

PL/I is available on five IBM platforms:

- MVS
- VM
- VSE
- Operating System/400 (*) (OS/400 (*))
- Operating System/2 (*) (OS/2 (*))

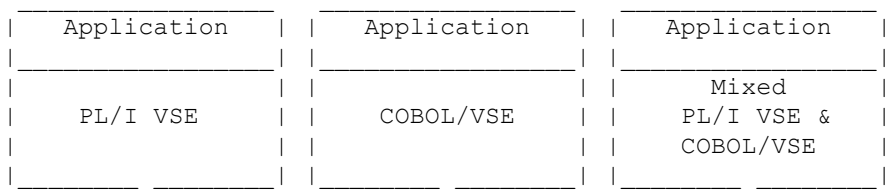
Because IBM implementations of PL/I are highly portable across these platforms, you spend less time training users and increase the efficiency of your data processing efforts.

PL/I VSE's relationship to LE/VSE

The MVS, VM, VSE, and OS/2 implementations of PL/I share a common run-time environment, Language Environment.

Language Environment consists of common conventions, run-time facilities, and callable services that provide a consistent application development environment.

In a single VSE application you can include modules created by any LE/VSE-conforming language (currently, PL/I VSE and COBOL/VSE). Figure 1 shows how LE/VSE provides a common run-time environment for VSE applications.



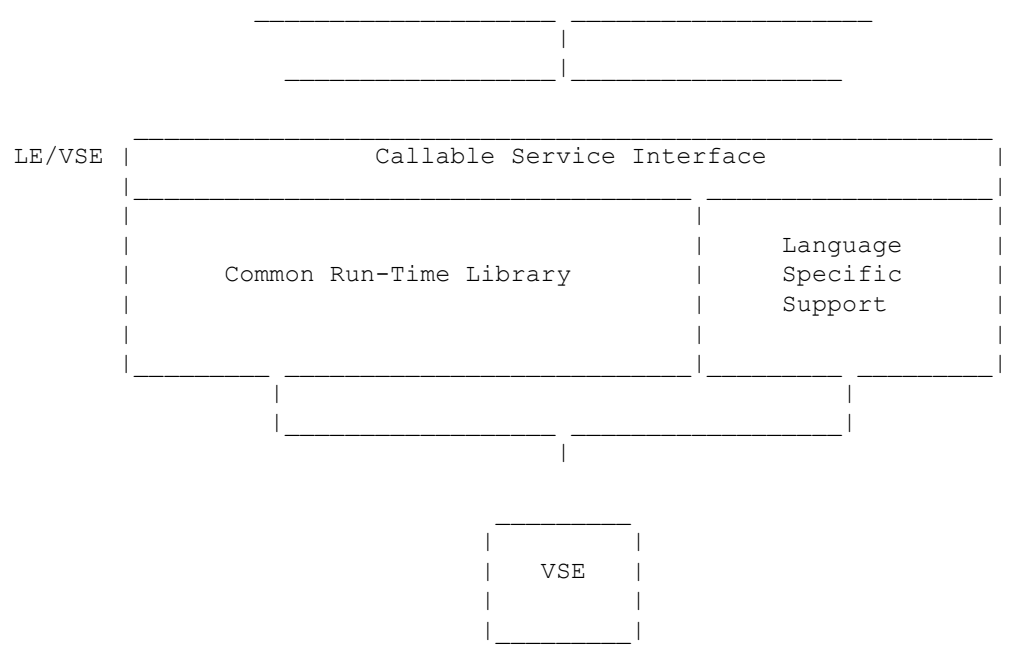


Figure 1. Developing VSE applications using PL/I VSE

2.0 How you can benefit from PL/I VSE

Take advantage of PL/I's language features

One benefit of PL/I is its block-oriented nature. A PL/I program consists of one or more blocks of statements. This block structure lets you produce modular programs of extensively reusable code because blocks can contain declarations that define variable names and control storage allocation. Thus, you can restrict the scope of a variable to a single block or a group of blocks, or can cause the variable to be known throughout the program.

By giving you the freedom to determine the degree to which a block is self-contained, PL/I makes it possible to produce blocks that other programs can freely use, but still allows interaction in blocks where interaction is desirable.

PL/I supplies many defaults. If you do not specify all attributes associated with a name or all options allowed in a statement, the compiler assigns attributes or options for you. This default action has two main advantages. First, it reduces the amount of declarations and other program statements that you must write. Second, it enables you to teach and use subsets of the language without being concerned about all possible alternatives.

Other PL/I language features include a wide variety of:

- Data types, including coded arithmetic, string, and several types for program control

- Data aggregates, including arrays, structures, and arrays of structures
- Built-in functions and subroutines
- Storage classes, including automatic, static, controlled, and based

Virtual Storage Constraint Relief (VSCR)

Although the PL/I VSE compiler runs below the 16-megabyte line, PL/I VSE applications created by the compiler can use VSE/ESA **extended architecture** 31-bit addressing. This allows you to take advantage of address space above the 16-megabyte line.

Thus, you can construct large applications that use extensive tables of data without resorting to techniques like segmentation to fit large programs into the available address space.

With VSCR:

- The operating system can load a PL/I VSE application at an address in virtual storage above the 16-megabyte line.
- The PL/I VSE application program can call programs that run above the 16-megabyte line--even if the calling program is loaded below 16 megabytes.
- The PL/I VSE application can access data that is loaded above or below the 16-megabyte line.

Improve programmer productivity

PL/I has several categories of built-in functions and subroutines that simplify many of your common programming tasks. These categories include math computation, integer manipulation, string-handling, precision specification, array processing, error handling, input and output procedures, and storage allocation.

Program interrupts or conditions that occur at run time can be detected by the hardware, the operating system, other software, or PL/I itself. Using the facilities available with PL/I, you can write applications that provide non-stop operation by handling these conditions.

In addition to condition handling, PL/I provides diagnostic facilities to prevent programming errors. For many of these errors, the running application can detect the problem, take corrective action, and continue execution. Problems that cause your programs to produce inaccurate

results include, for example:

- Subscripts or substrings that are out of range
- Assignment of a string to a target shorter than its source
- Computational results that exceed declared precisions

Some of the new features of this release of PL/I are:

- Enhanced interlanguage communication (ILC) with COBOL/VSE, including FETCH.
- Dynamic loading of subprograms using the FETCH statement.
- FETCHABLE procedures to simplify your selection of a procedure within a fetched phase.
- Automatic storage above the 16-megabyte line under control of the LE/VSE STACK option.
- The LE/VSE STORAGE option, which you can use to initialize storage, detect the use of uninitialized variables, and detect the use of free or never-allocated storage.
- OPTIONS(BYVALUE) and OPTIONS(BYADDR), which allow you to choose whether arguments are passed directly by value or indirectly by address.
- The NOT and OR compile-time options, which allow you to specify up to seven alternate symbols, any one of which may be used as:
  - The logical NOT operator, or
  - The logical OR operator and the string concatenation operator

  The default symbols for these operators are troublesome because upload and download programs often do not translate them correctly. Therefore, your ability to specify alternate symbols helps to make your source code more portable.
- OPTIONAL arguments, which allow you to omit arguments to assembler routines.
- The option to choose built-in mathematical functions that are compatible with either LE/VSE or DOS PL/I.

- Alternate external names for assembler entry points.
- The ability to customize dates and times.
- DATE and DATETIME functions that return millisecond resolution.
- Access to LE/VSE functions that properly handle 2-digit and 4-digit years in the year 2000 and beyond.

Enhanced interlanguage communication with COBOL/VSE

Your PL/I VSE applications can communicate with applications created by the COBOL/VSE compiler.

Using LE/VSE, your PL/I VSE applications can fetch COBOL/VSE routines; and COBOL/VSE applications can dynamically call PL/I VSE routines.

PL/I VSE now supports reentrant ILC applications.

ILC between PL/I VSE and COBOL/VSE is also supported under CICS/VSE (*).

Use powerful input and output facilities

PL/I's input and output facilities provide a high degree of application portability. These facilities also allow you to take advantage of data files, access methods, and other devices that the operating system provides.

PL/I handles most ordinary operations, such as file opening and closing. Programs can control input and output operations (such as checking on file status) as well as intercept various exceptional conditions to take corrective action. Such conditions include:

- End of file
- End of report page
- Record not found
- Unknown name in data-directed input
- Record too long or too short
- File not found

PL/I supports two types of data transmission. *Record-oriented* input and output transmits data aggregates, or records, one at a time without performing any data conversions. *Stream-oriented* input and output transmits one data item at a time in a continuous stream of characters. Data is converted during transmission--external data is in character format, and internal data is represented in any of the computational data types supported by PL/I.

Make your source programs device independent

PL/I VSE builds DTFs dynamically at run time. This means that you can define input and output device types in run-time JCL, making your source programs device independent.

Compatibility with DOS PL/I and PL/I MVS & VM

PL/I VSE provides source program compatibility with DOS PL/I and PL/I MVS & VM for most PL/I language elements.

Where there are differences between DOS PL/I and PL/I MVS & VM, PL/I VSE retains compatibility with DOS PL/I.

PL/I VSE does not provide object compatibility with either DOS PL/I or PL/I MVS & VM. To use the functions of PL/I VSE, application programs must be recompiled with the new compiler and link-edited with the LE/VSE run-time library.

3.0 Installing and customizing PL/I VSE

You can install and customize PL/I VSE by following the instructions in *PL/I VSE Installation and Customization Guide.* Note that if you have not already installed LE/VSE, you need to do so before installing PL/I VSE.

4.0 Software requirements checklist

PL/I VSE runs under the control of, or in conjunction with, the following licensed programs and their subsequent releases unless otherwise announced by IBM.

Operating system

One of the following:

- VSE/ESA Version 1 Release 4
- VSE/ESA Version 2 Release 1

Run-time environment

- IBM Language Environment for VSE/ESA Release 1

The PL/I VSE compiler requires LE/VSE for compilation, and PL/I VSE programs require LE/VSE for linking and execution.

The following program is required for customizing compiler options:

- High Level Assembler/MVS & VM & VSE Release 1

Related IBM products supported by LE/VSE and PL/I VSE

- IBM COBOL for VSE/ESA Release 1
- CICS/VSE Version 2 Release 3
- DFSORT/VSE Version 3 Release 1
- DOS/VS VM/SP Sort/Merge Version 2 Release 5
- DL/I DOS/VS Release 10
- SQL/DS (*) Version 3 Release 4

5.0 Hardware requirements

PL/I VSE object code will run on any hardware configuration supported by the licensed programs specified above.

6.0 Ordering PL/I VSE

For more information and ordering materials, call your IBM marketing representative.

IBM

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service.

Any other documentation with

respect to this licensed
program, including any
documentation referenced herein,
is provided for reference
purposes only and does not
extend or modify these
specifications.

Printed in U.S.A.

IBM Library Server Print Preview

DOCNUM = GC26-8052-00
DATETIME = 02/07/95 11:10:48
BLDVERS = 1.2
TITLE = PL/I VSE Fact Sheet
AUTHOR =

PATH = /home/webapps/epubs/htdocs/book
