Reduce mainframe development costs and increase productivity using PL/I on your workstation

IBM SAA AD/Cycle PL/I Package/2

Overview

IBM* SAA* AD/Cycle* PL/I Package/2 offers a number of benefits to new and existing PL/I users.

As an existing PL/I user, you can check programs for syntax and semantic errors on the workstation and compile and run them on the mainframe. As you relieve your mainframe of a significant portion of its heavy workload, development and production costs decrease.

As a new or existing PL/I user, this new product allows you to take full advantage of productivity tools and features available to workstation programmers. Among these features are an isolated test and development environment, and the ability to develop standalone Presentation Manager* applications.

PL/I is a powerful programming language designed to provide solutions for your application development needs.

For customers with large libraries of PL/I code, PL/I Package/2 helps you protect your investment and continue to use your base of trained PL/I developers in a more efficient way. For potential customers, PL/I Package/2 offers a wide variety of language features to effectively create a wide variety of applications, such as:

- String handling for business applications
- Floating point operations for science and engineering applications
- Pointer arithmetic for system programming applications.

With the interface provided by OS/2*, you can simultaneously work within multiple windows containing different development activities. This multitasking, graphical operating system provides a simple and highly productive development environment.

PL/I Package/2 addresses numerous user requirements submitted by SHARE and GUIDE. It also supports most of the American National Standard PL/I General Purpose Subset (ANSI X3.74-1987) and implements a significant number of extensions that allow you to take advantage of OS/2 functions and facilities.

Meeting Your Application Development Needs

Today's enterprises need efficient, consistent, and simple ways to develop quality applications and maintain their 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.

The SAA Strategy

Systems Application Architecture (SAA) is a collection of selected software interfaces, conventions, and protocols designed to help you develop consistent applications across current and future offerings of four major IBM computing environments:

- MVS
- VM with CMS
- Operating System/400* (OS/400*)
- Operating System/2* (OS/2).

Programs that run on each of the systems have similar interfaces. This reduces the amount of time needed to train users and increases the efficiency of your data processing efforts.

PL/I's Position in the AD/Cycle Framework

AD/Cycle is part of the Systems Application Architecture strategy intended to solve the problem of efficiently developing and maintaining applications. The AD/Cycle framework divides the application development process into five phases. PL/I Package/2 falls under the "Languages" category of this development cycle. Included with PL/I in this category are compilers for other OS/2 languages. Also included is a common run-time environment, Language Environment*, to support those languages collectively (see Figure 1).

Language Environment consists of common conventions, run-time facilities, and callable services that provide a uniform and consistent application development environment. Using languages that are members of the Language Environment group or other languages that conform to certain linkage conventions, you can include modules from various languages in a single application. PL/I Package/2 has the first OS/2 language compiler to utilize this common run-time environment.

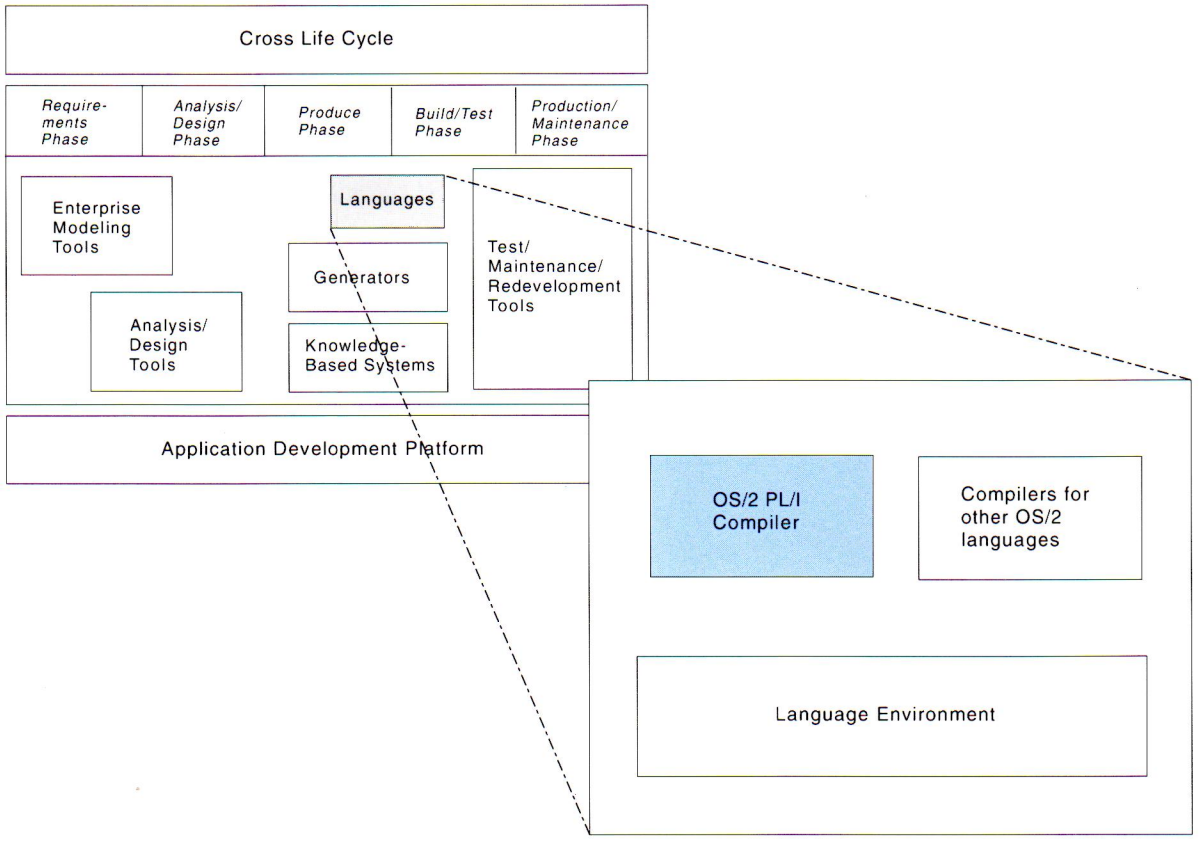

Figure 1. How PL/I Package/2 Fits within the AD/Cycle Framework

Benefitting from PL/I Package/2

PL/I Package/2 supplies the features and functions that improve the efficiency of your application development processes—now and in the future.

Take Advantage of PL/I's Language Features

One benefit of PL/I is its block-oriented nature. Structure elements within the language allow for extensive code reuse. The PACKAGE statement, for example, provides:

- A structured way to group related declarations and procedures that share name scope
- Controlled exposure of static variables and external procedures
- The ability to create data-only modules
- An alternative to multiple entry points that makes your programs more readable and maintainable.

PL/I has no reserved keywords. This lets you choose identifier names without having to worry that these names might be restricted as keywords.

Restricted expressions and named constants allow you to create parameters in your source code so that when one item changes, all related items automatically adjust. These expressions are evaluated at compile time, reducing run-time cost.

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

- Storage classes, including: automatic, static, controlled, and based
- Interprocedure linkages, including all 32-bit linkages
- Data types, including unsigned fixed binary
- Data aggregates, including: arrays, structures, and unions.

Improve Programmer Productivity

A significant number of new functions and subroutines are built into PL/I, bringing the total to nearly 200.

Brief descriptions of selected new built-in functions follow:

String-handling Built-in Functions

Function	Description
CENTER	Returns a string with a value centered in it
COPY	Returns a string consisting of n copies of a string
LEFT	Returns a string with a value left justified in it
REVERSE	Returns a reversed image of a string
RIGHT	Returns a string with a value right justified in it
SEARCH	Searches for the first occurrence of any one of the elements of a string within another string
TRIM	Trims specified characters from the left and right sides of a string

Integer Manipulation Built-in Functions

Function	Description
IAND	Bitwise "and" of 2 fixed binary values
IEOR	Bitwise "exclusive-or" of 2 fixed binary values
IOR	Bitwise "or" of 2 fixed binary values
LOWER2	Shifts a fixed binary value algebraically left
RAISE2	Shifts a fixed binary value algebraically right

Miscellaneous Built-in Functions

Function	Description
COLLATE	Returns a character(256) string specifying the collating order
COMPARE	Compares n bytes at two addresses
HEX	Returns the hex representation of a value
HEXIMAGE	Returns the hex representation of a specified number of bytes at a given address
OMITTED	Indicates if a parameter was not supplied
VALID	Indicates if the contents of a fixed decimal or picture variable are valid
PLIFILL	Fills n bytes at an address with a specified byte value
PLIMOVE	Moves n bytes from one address to another

These functions and subroutines simplify many common programming tasks. Additional categories of built-in functions include: math computation, precision specification, array processing, error handling, input and output procedures, and storage allocation.

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

In addition to condition handling, PL/I provides diagnostic facilities to prevent programming errors. Problems that cause your programs to produce inaccurate results include the following:

- 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.

For all of these (and other) potential errors, the running application can detect the problem, take corrective action, and continue execution.

Tailor PL/I Package/2 for Your Organization

The PL/I Package/2 compiler supports an exit routine that you can use to alter the severity of compiler messages or suppress them completely.

To help you in debugging your program, the compiler creates a listing of comprehensive diagnostic messages that identify errors in the source program. Depending on what compiler options you select, run-time messages identify the number of the statement in error.

You can choose to have an attribute and cross reference table included in your program listing. This table not only tells you in which lines variables are referenced, but also in which lines variables are altered.

New Compiler Options

New compiler options provide valuable function to the PL/I compiler, for example:

- The DEFAULT option allows you to change defaults for attributes and options at compile time without having to change your source code. For example, you can choose:
  - Whether arguments are passed by value or by address.
  - If parameters reference connected or non-connected storage.
- The LANGLVL option lets you decide whether or not the compiler flags your use of features not included in the SAA definition of PL/I
- The LIMITS option allows a fixed decimal precision of 31.
- The NAMES option lets you specify a series of characters that the compiler accepts in identifiers. You can, therefore, create names using your own national characters.
- The NOT and OR compiler options let you replace the not (¬) and or (|) symbols with alternate code points.
- The OPTIMIZE option lets you choose between faster compilation and the generation of more efficient code.
- The PREFIX option allows user-selected condition prefixes to be applied dynamically, without requiring the user to edit the source code.
- The RULES option lets you decide whether or not implicit declarations are allowed.

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.

The PL/I Package/2 compiler 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 data as a continuous stream of characters, sending one data item at a time. 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.

Installing PL/I Package/2

To install PL/I Package/2, use the automated installation process provided with the product (documented in the PL/I Package/2 installation information). This process eliminates the need to manually complete tasks such as updating OS/2's CONFIG.SYS. Automated installation is available for both installation from diskettes and installation from tape.

System Configuration

PL/I Package/2 requires a PS/2® (or true compatible) with at least an 80386SX processor. An 80387 (or compatible) math coprocessor is required unless your machine has an 80486 processor (which includes a built-in math coprocessor). The minimum operating system software requirement is OS/2 Version 2.0 (or later).

In addition, the following hardware is recommended:

- 8M byte RAM
- 60M byte hard disk, with 10M bytes of free space.

As is true with most software, the performance of PL/I Package/2 generally improves with increased RAM and hard disk capacity.

Virtual Storage

With PL/I, compilation of a typical 200-statement program requires about 5M bytes of virtual storage. Larger and more complex programs require more virtual storage.

Application Requirements

Storage requirements for a PL/I application depend on the size of the application and nature of the logic. In addition to program storage, 2M to 3M bytes are required for run-time support.

Ordering PL/I Package/2

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

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.

September 1992

Printed in U.S.A.

GC26-3090-00