



interskill
learning

Introduction to Language Environment

By proceeding with this courseware you agree with [these terms and conditions](#). Interskill Learning Pty. Ltd. © 2019





Objectives

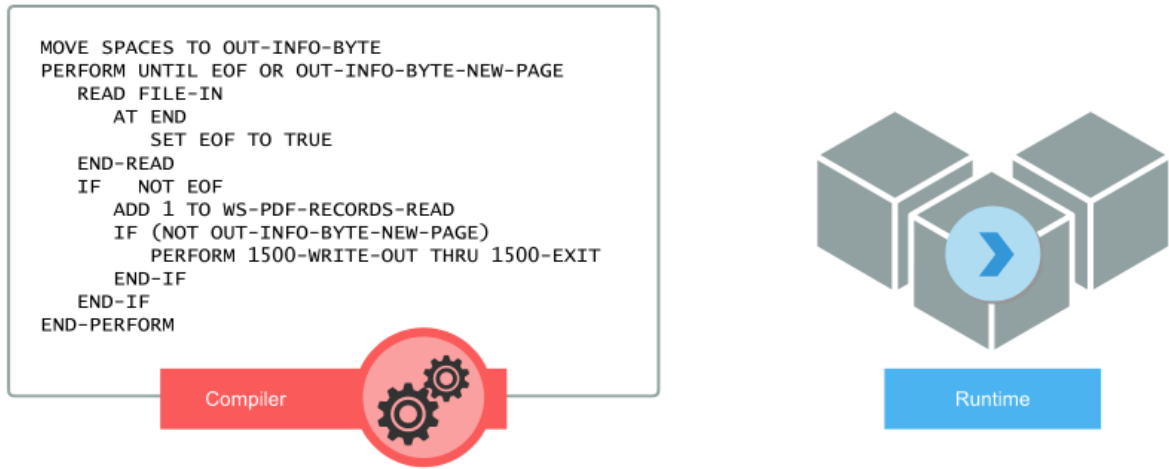
Introduction to Language Environment

This module will introduce you to the IBM Language Environment, or LE.

The z/OS component Language Environment is critical for all IBM High Level compiled languages, including COBOL, PL/1, C, C++, and Fortran. Anyone working with these languages, especially application programmers will need to know about Language Environment.

After completing this module, you should be able to:

- Describe the Purpose of Language Environment
- Understand How Language Environment Error Conditions are Handled



Compiled languages such as COBOL, C, and PL/1 need two components. The compiler, which is a program to create executable code from the source, and the runtime, which is a set of module used by the program during execution.



Initialization



Message Handling



Storage Management



Termination



Error Handling and Recovery



Common Routines

The runtime environment, or runtime libraries, perform many different functions. Without runtime libraries, applications programs would each have to work with the z/OS operating system to perform things like initialization, memory handling, error recovery, and more.

Shown here is a list of features included in most runtime libraries. **Mouse-over** each to find out more.



Initialization



Message Handling



Storage Management



Termination



Error Handling and Recovery

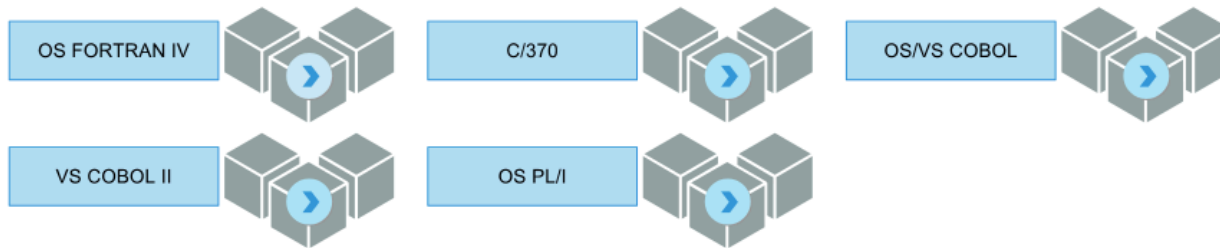


Common Routines

Runtime libraries gain control when a compiled program is called, and prepare the environment for the program. They obtain working storage, prepare parameters, and perform other tasks before a compiled program can start. If this program is called by another, the calling program's environment is saved.

The runtime environment, or runtime libraries, perform many different functions. Without runtime libraries, applications programs would each have to work with the z/OS operating system to perform things like initialization, memory handling, error recovery, and more.

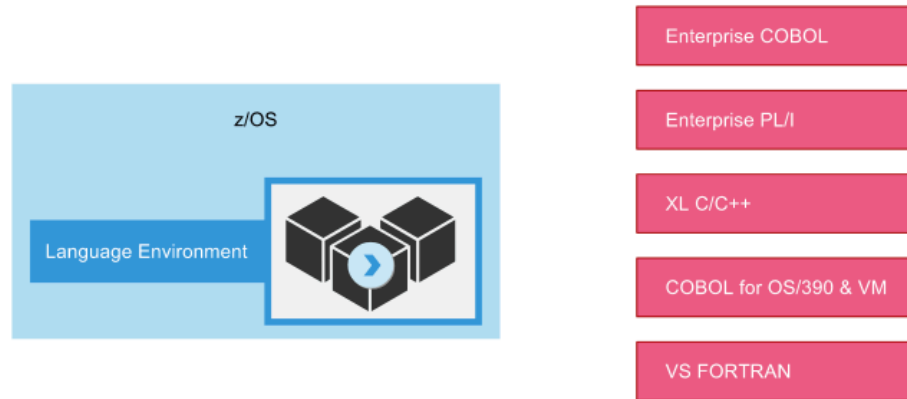
Shown here is a list of features included in most runtime libraries. **Mouse-over** each to find out more.



- >> COBOL Calling PL/I
- >> PL/I Calling FORTRAN
- >> C Calling COBOL
- >> Common Routines for COBOL and C

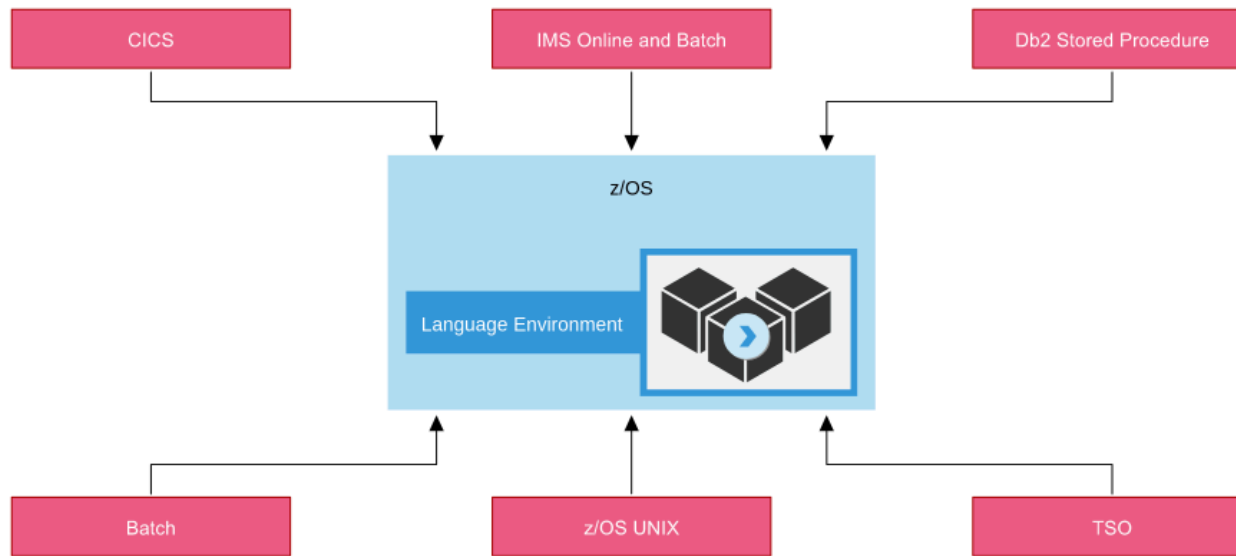
In the past, each compiler had separate runtime libraries. These were often incompatible with each other. It was difficult for a COBOL program to call a PL/I program, or to develop a common routine that could be called from different languages.

It also opened possible problems with multiple runtime libraries attempting to run at the same time on z/OS.

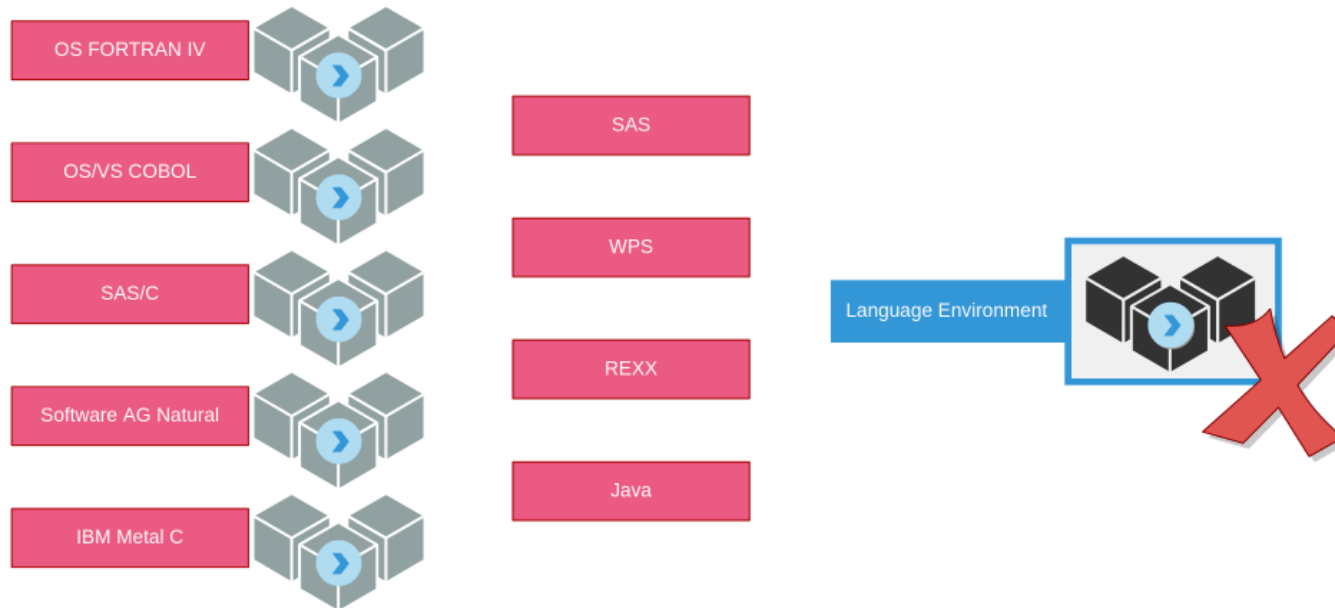


To solve this problem, IBM created common runtime libraries to use with all IBM compiled programming languages: Language Environment (LE). LE is a component of z/OS, and is included free of charge.

Today all current IBM compiled languages require and use LE.

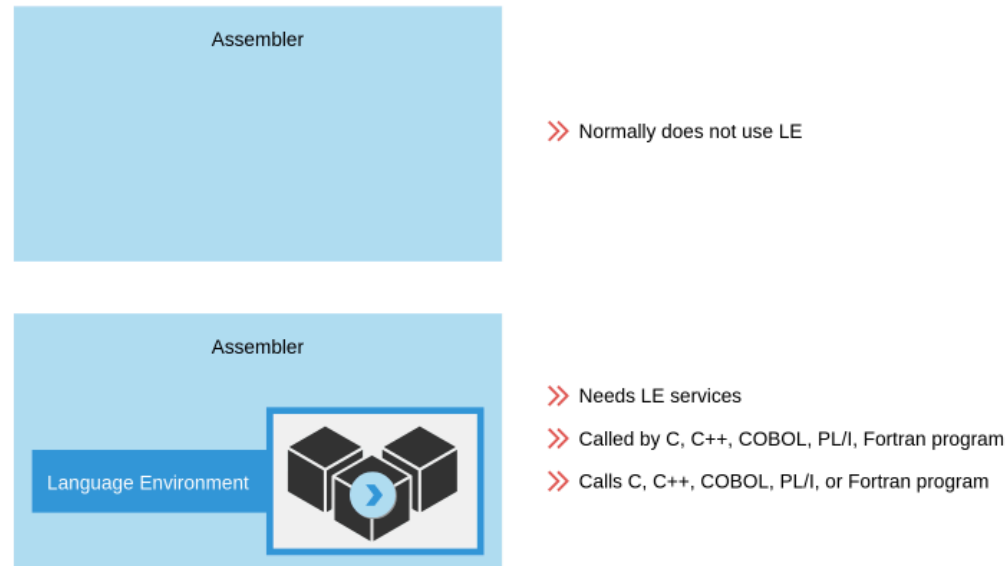


Regardless of where a program executes - in batch, CICS, IMS, UNIX, or TSO, current IBM compiled programs always use Language Environment.



However not all programming languages use Language Environment. Languages that do not use Language Environment include the following:

- Non-compiled IBM languages such as REXX, Java, PHP, and CLIST.
- Non-compiled languages from other vendors such as SAS, CA Easytrieve, and World Programming WPS.
- Older IBM compiled languages such as OS/VS COBOL, VS COBOL II, Ada/370, APL2, and Pascal.
- Compiled languages from other software vendors such as SAS/C and Software AG Natural - these have their own runtime libraries.
- IBM Metal C - a module produced by IBM XL/C that is designed to be used without LE.



Assembler programs often run independently of a runtime environment, and do not need to use Language Environment. However, Language Environment provides macros to allow Assembler programs to use Language Environment. These are called LE-compliant assembler modules. LE-compliant assembler may be used under the following conditions:

- An Assembler program wants to use Language Environment features
- An Assembler program will call a High Level Language program using LE
- An Assembler program will be called by a High Level Language program using LE