



CICS Transaction Server for z/OS

Language Environment within CICS TS:  
Questions and Answers

Robert Harris  
CICS Technical Strategy  
IBM Hursley

Issued:	01 October	2002
Revision Date:	01 November	2003
Previous Revision Date:	01 October	2002
Review Date:		As required

**Take Note!**

Before using this document be sure to read the general information under "Notices".

Second Edition, November 2003.

© **Copyright International Business Machines Corporation 2003.** All rights reserved.  
Note to US Government Users -- Documentation related to restricted rights -- Use,  
duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule contract  
with IBM Corp.

**Notices:**

The following paragraph does not apply in any country where such provisions are inconsistent with local law.

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

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 licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program that does not infringe any of the intellectual property rights may be used instead of the IBM product.

Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents.

You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, New York 10594, USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS-IS. The use of the information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

## Trademarks:

The following are Trademarks of International Business Machines Corporation in the United States, in other countries, or both:

3090	ACF/VTAM	AD/Cycle
AFP	AIX	AnyNet
Application System/400	APPN	AS/400
AT	BookManager	C Set++
C/370	C/MVS	CBIPO
CBPDO	CICS	CICS/400
CICS/6000	CICS/ESA	CICS/MVS
CICS OS/2	CICS TS	CICS/VM
CICS/VSE	CICSplex	CICSplex SM
COBOL/370	COBOL/2	Common User Access
CUA	DATABASE 2	DB2
DFSMS	DFSMS/MVS	DFSMSdfp
DFSMSdss	DFSMShsm	DFSORT
DXT	eNetwork	Enterprise Systems Architecture/370
Enterprise Systems Architecture/390	ES/3090	ESA/370
ESA/390	ES/9000	ESCON
GDDM	HiperBatch	Hiperspace
InfoWindow	IBM	IBMLink
IMS	IMS/ESA	Language Environment
MQ	MQSeries	MVS
MVS/DFP	MVS/ESA	MVS Parallel Sysplex
MVS/SP	MVS/XA	Multiprise
NetView	OpenEdition	OS/2
OS/390	OS/400	Processor Resource/Systems Manager
Parallel Sysplex	PR/SM	Presentation Manager
RACF	Resource Measurement Facility	RETAIN
RISC System/6000	RMF	RT
S/370	S/390	SAA
SQL/DS	SP	System/36
System/38	System/360	System/370
System/390	SystemView	Systems Application Architecture
VisualAge	VSE/ESA	VTAM
WebExplorer	z/OS	

UNIX is a registered Trademark in the United States and other countries licensed exclusively through X/Open Company Limited .

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

INTEL is a registered trademark of Intel Corporation, in the United States, or other countries, or both.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, or other countries, or both.

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

### **Summary of amendments:**

Date Of Change	Change made to document
01/10/2002	Creation
01/11/2003	Updated for CICS Transaction Server V2.3

### **Reference Material and Bibliography:**

In addition to the general CICS TS and Language Environment documentation, these manuals are especially relevant:

SC28-1944-03	Language Environment for z/OS and OS/390 Run-Time Migration Guide
GC27-1409-00	Enterprise COBOL for z/OS and OS/390 Compiler and Run-Time Migration Guide
GC27-1458-00	Enterprise PL/I for z/OS and OS/390 Compiler and Run-Time Migration Guide
GC09-4913-00	C/C++ Compiler for z/OS and OS/390 Compiler and Run-Time Migration Guide
SA22-7564-02	z/OS Language Environment Customization

## **Preface:**

This document is directed at CICS System Programmers who want to understand the implications of installing Language Environment into a CICS TS region.

It is aimed at taking a System Programmer who is knowledgeable about the CICS TS environment through a series of Questions and Answers about the installation of Language Environment and the associated implications for OS/VS COBOL, VS COBOL II, OS PL/I and OS C programs.

Knowledge of Java™, Enterprise Java Bean™ (EJB™) or Enterprise Bean™ function in the CICS Transaction Server for z/OS environment is not required.

The information in this document is only applicable to CICS Transaction Server for z/OS Version 1.3 and Version 2.3. It does not apply to other CICS releases. The notation **CICS TS >2.3** means the next release of CICS Transaction Server after Version 2.3.

Included in this edition are summary tables showing the support status for compiled languages within CICS TS 2.3.

The contents of this document should be treated as **advice only**. Full and detailed information is contained within the documentation for CICS TS and Language Environment.

## Table of Contents

Introduction and definitions .....	1
Language Environment Considerations for CICS TS 1.3 and CICS TS 2.3 regions.....	3
Language Environment runtime installation issues .....	7
Considerations for OS/VS COBOL programs .....	11
Considerations for VS COBOL II programs.....	15
Considerations for VS COBOL II programs.....	15
Considerations for PL/I programs .....	22
Considerations for C/370 programs .....	27
CICS Translator support.....	32
CICS runtime support for unsupported native libraries .....	33
CICS runtime support for supported native libraries .....	34
CICS support for the Language Environment runtime .....	35
CICS Integrated Translator support .....	36
Use of the IBM Distributed Debugger .....	37
Use with WebSphere Studio Enterprise Developer .....	38

## Introduction and definitions

This document considers the effect on CICS applications running within the CICS TS environment when Language Environment is placed within the CICS TS region. It consists of a series of Questions and Answers aimed at the CICS System Programmer.

Some confusion has arisen over the exact meaning of 'Installing Language Environment within CICS'. This term has been laxly used. It covers several arenas, all of which are separate:

1. It can mean using runtime libraries provided as part of the z/OS Language Environment component
2. It can mean using a runtime supported by IBM for CICS application programs (such as a runtime for OS/VS COBOL modules)
3. It can mean running CICS application programs using the z/OS Language Environment libraries

Confusion has arisen because of the requirement in the release of CICS TS Version 2.3 to use the Language Environment runtime (point number 3).

If the Language Environment runtime is not being used in CICS TS 1.3 or CICS TS 2.2 (where it is optional) applications may notice an effect when changing from an unsupported runtime (point number 2) to the Language Environment runtime required in CICS TS 2.3. The application module will not have changed, but as the Language Environment runtime is controlled by options, these settings may introduce a changed behaviour. An example of this is to do with the manner in which obtained storage is initialised. This type of change is briefly considered in this document.

CICS TS 2.3 will support the execution of OS/VS COBOL application program modules provided the OS/VS COBOL runtime is supported by IBM. This supported runtime is provided within the Language Environment libraries (point number 1).

**However, CICS TS 2.3 will be the last release of CICS to support the execution of OS/VS COBOL modules.**

Thus, CICS TS 2.3 (as well as CICS TS 1.3 & CICS TS 2.2) will run OS/VS COBOL modules in a Language Environment provided runtime which is distinct from the Language Environment runtime used for other modules. The implications of this are discussed in this document.



A discussion about these issues from the standpoint of a CICS TS 1.3 (or CICS TS 2.2) region not containing a Language Environment is on page 3. Issues relating to the Language Environment runtime are documented on page 7.

A discussion about the implication for modules compiled with an OS/VS COBOL compiler is on page 11.

Modules compiled with VS COBOL II, early versions of PL/I or early versions of C can run with either their own runtime or within the Language Environment runtime. Implications for these modules are discussed on page 15 for VS COBOL II, page 22 for PL/I and page 27 for C.

The runtime for an application module within a CICS TS region is a choice between the native runtime suitable for that module and the Language Environment runtime. This topic is discussed with regard to CICS TS 1.3 and CICS TS 2.3.

This document aims to be a simple Question and Answer to inform the reader. A reader with expert knowledge of Language Environment processing and CICS TS functionality will be able to discern additional options. However, as these are not recommended for general use, they are not discussed in this text. After the Q&A sections, summary tables show support status and usage for high-level compiled languages with CICS – this information starts on page 32.

The author assumes the progress of CICS TS from Version 1.3 to Version 2.3 omitting CICS TS Version 2.1 and CICS TS Version 2.2. The recommended migration for customers using CICS TS Version 1.3 is to CICS TS Version 2.3.

Advice given in this document is relevant to the great majority of application programs and modules. However, there are some cases where advice about not recompiling or relinking a module may be inaccurate for a very small number of highly complex programs using very old compilers.

Consequently, this document is only for general guidance. Full and definitive information is contained within the CICS TS and Language Environment documentation.

If you need to raise a problem with IBM associated with the issues discussed in this document, you must be using a supported runtime.

# Language Environment Considerations for CICS TS 1.3 and CICS TS 2.3 regions

This section discusses Language Environment function from the viewpoint of a CICS TS region without a supported runtime moving to a supported runtime. Language related issues are discussed subsequently, starting on page 11.

If the CICS TS region already contains a Language Environment runtime, most of the Questions in this section will already have been considered.

## CICS TS 1.3 and CICS TS 2.3

**Q** How does CICS run OS/VS COBOL modules?

**A** CICS determines that the module is OS/VS COBOL and generates a unique environment for that module using a specific runtime. This environment is not the same as the Language Environment runtime environment, and they are logically different.

**Q** What is the supported runtime for OS/VS COBOL programs?

**A** That provided within the Language Environment libraries (which must be placed in the JCL or be available via LNKLIST). This does not mean you have to install the Language Environment runtime within the CICS region. However, if you have modules compiled with other languages, you will require the Language Environment runtime.

- |  |   |
|--|---|
| <p>Q How does CICS run VS COBOL II modules?</p>  | <p>A It depends upon whether or not the Language Environment runtime is present.<br/>If CICS determines that the module is VS COBOL II and the Language Environment runtime is present, then the VS COBOL II module will run within the Language Environment runtime.<br/>If the Language Environment runtime is not present, then the VS COBOL II runtime (which is unsupported) will be used for the module (if this VS COBOL II runtime is absent, the module will not be executable).<br/>In CICS TS 1.3 the Language Environment runtime is optional: in CICS TS 2.3 it is required.</p> |
| <p>Q Must I have to have the Language Environment runtime within my region?</p>  | <p>A Yes: It is required by CICS TS 2.3.</p>  |
| <p>Q What is the effect on OS/VS COBOL modules when I install the Language Environment runtime within my CICS region?</p>  | <p>A None. However, you get the advantage of using a supported runtime.</p>   |
| <p>Q Does this mean I can use an unsupported OS/VS COBOL runtime at the same time as the supported Language Environment runtime (for VS COBOL II programs etc.)?</p> | <p>A Yes: But this is not recommended. You can arrange this by putting the unsupported OS/VS COBOL runtime library higher up in the DFHRPL &amp; STEPLIB/LNKLST concatenations than the Language Environment runtime library.</p>   |
| <p>Q Can I use LNKLST for Language Environment runtime libraries?</p>  | <p>A Yes: However, using LNKLST may force the use of the Language Environment runtime within the CICS TS region whereas using STEPLIB and DFHRPL is more specific.</p>  |
| <p>Q Are there any virtual storage implications for the CICS TS region when the Language Environment runtime is installed?</p>                                       | <p>A There are implications for the CICS TS region in both the way virtual storage is used and the quantity required. The migration guides discuss these issues, but CICS' EDSALIM will most likely have to increase.</p>   |

<p>Q How do I change to a supported runtime for OS/VS COBOL modules?</p>	<p>A Use the Language Environment libraries in STEPLIB/LNKST &amp; DFHRPL concatenations and remove the unsupported runtime from the DFHRPL &amp; STEPLIB/LNKST concatenations. The supported OS/VS COBOL runtime is contained within the Language Environment library (SCEERUN).</p> <p>In the unlikely case if you only have OS/VS COBOL and assembler modules in the CICS TS region, or you just want to use a supported library for OS/VS COBOL modules only, then you do not need to RDO define the Language Environment group (if you do this, you will enable the Language Environment runtime within the CICS region for other modules). In most cases, however, you will need the RDO group installed.</p>
<p>Q Do I have to recompile my OS/VS COBOL source?</p>	<p>A No.</p>
<p>Q Do I have to relink my OS/VS COBOL module</p>	<p>A No.</p>
<p>Q I'm going to run a VS COBOL II module within the Language Environment runtime. Do I have to recompile?</p>	<p>A No.</p>
<p>Q I'm going to run a VS COBOL II module within the Language Environment runtime. Do I have to relink?</p>	<p>A No: But the execution of the VS COBOL II module will depend on Language Environment settings which may mean a relink is required.</p>
<p>Q I'm going to use the Language Environment runtime instead of the PL/I or C runtime. Do I have to recompile?</p>	<p>A No.</p>
<p>Q I'm going to use the Language Environment runtime instead of the PL/I or C runtime. Do I have to relink?</p>	<p>A No: But the execution of the PL/I or C module will depend on Language Environment settings which may mean a relink is required. However, there is less of an impact than for VS COBOL II.</p>

Q Can I run an OS/VS COBOL module using an unsupported runtime?

A Yes: They will run without the need to recompile or relink. But you should not be using an unsupported runtime.

Q How can I have modules using an existing unsupported runtime whilst running other modules within the Language Environment runtime?

A With the exception of OS/VS COBOL modules that operate within their own runtime, all other modules will use the Language Environment runtime. Consequently, you cannot have some modules using the Language Environment runtime and some using an unsupported runtime.

Q But I don't want to migrate all my modules in a given CICS TS region to the Language Environment runtime all at once...?

A In this case you are going to have to separate modules into different CICS TS regions: one region has the Language Environment runtime whilst the other only contains the unsupported runtimes. However, this can only be done for CICS TS 1.3 as CICS TS 2.3 requires the Language Environment runtime.

## Language Environment runtime installation issues

This section discusses issues relating to the installation of the Language Environment runtime into a CICS region. Although the Language Environment runtime is optional for CICS TS 1.3, it is required for CICS TS 2.3.

If you have already installed the Language Environment runtime, most of the issues in this section will already have been resolved. Otherwise, it is recommended that the Language Environment runtime be installed as early as possible to avoid these issues occurring at a later date. It is important that all available Language Environment fixes are active for CICS TS 2.3.

Language related issues are discussed subsequently, starting on page 11.

### Installing the Language Environment runtime in a CICS TS 1.3 or 2.3 region

- |  |  |
|--|--|
| <p>Q Which libraries do I need to insert into the DFHRPL and STEPLIB/LNKLST concatenations to use the Language Environment runtime within CICS TS 1.3?</p> | <p>A You must insert the SCEECICS and SCEERUN libraries into the DFHRPL and STEPLIB/LNKLST concatenations in place of the existing unsupported language runtimes.</p>  |
| <p>Q Which libraries do I need to insert into the DFHRPL and STEPLIB/LNKLST concatenations to use the Language Environment runtime within CICS TS 2.3?</p> | <p>A You must insert the SCEECICS, SCEERUN and SCEERUN2 libraries into the DFHRPL and STEPLIB/LNKLST concatenations in place of any existing unsupported language runtimes.</p>  |
| <p>Q What changes need to be made for Language Environment between CICS TS 1.3 and CICS TS 2.3?</p>  | <p>You must use the latest versions of the SCEECICS and SCEERUN libraries provided by z/OS and ensure that the SCEERUN2 library is accessed. Ensure that the CEE group in the CSD is updated with the latest definitions.</p>    |
| <p>Q What do I need to do for RDO?</p>   | <p>A You must install the Language Environment RDO group (CEE). This is shipped within the SCEESAMP file as member CEECCSD. These definitions will be inserted into the CSD as part of the CICS TS 2.3 installation process.</p> |

<p>Q How do I configure Language Environment processing for CICS?</p>	<p>A As part of Language Environment installation the CEECOPT table containing Language Environment runtime options control operations within all CICS TS regions. One can optionally compile a default parameter table (CEEROPT) and place it in a load library within the DFHRPL concatenation used by the CICS region to control matters for a specific region.</p> <p>These issues are discussed in SA22-7564-02: z/OS Language Environment customization.</p>
<p>Q For a specific region?</p>	<p>A You have to compile distinct tables and put them in different non-shared load libraries so that each region can have it's own (CEEROPT) settings.</p>
<p>Q Do Language Environment runtime options used for z/OS batch processing affect modules running within the CICS TS environment?</p>	<p>A No: Batch settings are controlled by the CEEDOPT table. Thus, Language Environment settings for CICS and Batch do not interact.</p>
<p>Q How do the Language Environment runtime options affect an OS/VS COBOL program?</p>	<p>A They have no effect. The supported OS/VS COBOL runtime is contained in the same library as that used for the Language Environment runtime, but they do not interact with each other.</p>
<p>Q How do the Language Environment runtime options affect the execution of a VS COBOL II module that was not previously executing with a Language Environment runtime?</p>	<p>A The Language Environment runtime options can have a significant impact on an executing VS COBOL II module within the Language Environment. An example of this is to do with the way obtained storage is initialised. The required Language Environment settings depend upon what the VS COBOL II program was assuming about its runtime behaviour.</p>



- |  |  |
|--|--|
| <p>Q How do the Language Environment runtime options affect the execution of an early PL/I module or an early C module, which were not previously executing with a Language Environment runtime?</p> | <p>A The Language Environment runtime options will affect the running of these modules, but the impact is less than for the VS COBOL II case (due to language syntax).</p>   |
| <p>Q How can I configure Language Environment processing for a group of programs like all the PL/I modules or all the VS COBOL II modules?</p>   | <p>A You cannot configure the Language Environment runtime for a category of modules (but some runtime options are specific to a language) The settings apply to everything throughout the region unless overridden by a specific module itself.</p> |
| <p>Q Can I dynamically set Language Environment runtime options for a given module?</p>  | <p>A Yes: You can use Language Environment exit programs (such as CEEBXITA) to do this very specific parameter setting, but this is not usually recommended.</p>   |
| <p>Q But I do need different settings for different categories of modules. How do I accomplish this?</p>   | <p>A In this case you will have to separate modules into their own CICS regions and apply a different set of regionwide Language Environment runtime options to each region.</p>   |
| <p>Q How do I set a specific set of Language Environment runtime options for a specific VS COBOL II module?</p>  | <p>A You must code a Language Environment parameter table (CEEUOPT) and link it into the VS COBOL II module.</p>   |
| <p>Q How do I get a specific set of Language Environment runtime options for a specific PL/I module?</p>   | <p>A You must code a Language Environment parameter table (CEEUOPT) and link it into the PL/I module. However, you can also supply Language Environment settings within the PL/I program by using the PLIXOPTS external variable.</p>                |
| <p>Q How do I get a specific set of Language Environment runtime options for a specific C module?</p>  | <p>A You must code a Language Environment parameter table (CEEUOPT) and link it into the C module.</p>   |



Q Does the use of the Language Environment runtime alter dump, trace or other diagnostic processing?

A The Language Environment runtime will use CICS facilities for this processing in a different fashion to the language-specific runtimes. New Transient Data Queue destinations may need to be defined by RDO.

Abend codes will change. Language Environment exit programs can be used to restore the language-specific Abend codes.

## Considerations for OS/VS COBOL programs

This section describes migration considerations from a CICS TS region without a supported runtime to a supported runtime for modules compiled by:

### OS/VS COBOL Compilers

OS/VS COBOL (5740-CB1)  
OS/VS COBOL (5734-CB4)  
OS/VS COBOL (5740-LM1)

Modules produced by these compilers can only execute within an OS/VS COBOL runtime. This section assumes that they are not currently using a supported runtime.

The supported runtime for OS/VS COBOL programs is contained within the Language Environment libraries. You do not have to use the Language Environment runtime just to use the supported OS/VS COBOL runtime. However, if the Language Environment runtime is present within the CICS region, the OS/VS COBOL runtime is distinct from it.

**CICS TS 2.3 will be the last release of CICS to support the execution of OS/VS COBOL modules.**

COBOL programs compiled with later compilers are discussed on page 15.

### OS/VS COBOL

**Q** How does CICS run OS/VS COBOL modules?

**A** CICS determines that the module is OS/VS COBOL and generates a unique runtime environment for that module. This OS/VS COBOL environment is not the same as a full Language Environment runtime environment, and they are logically distinct.

**Q** Can I run an OS/VS COBOL module using an unsupported runtime within CICS TS 1.3 or CICS TS 2.3?

**A** Yes: They will run without the need to recompile or relink. But you should not be using an unsupported runtime and there is no advantage to so doing.

**Q** Should I run an OS/VS COBOL using an unsupported runtime within CICS TS 2.3?

**A** No: You should not be using an unsupported runtime and there is no advantage in so doing. You will already have the required libraries in DFHRPL and STEPLIB/LNKLST to use the supported runtime.

Implications for CICS TS modules when Language Environment is installed

Considerations for OS/VS COBOL programs

<p>Q How do I change to using a supported OS/VS COBOL runtime for CICS TS 1.3?</p>	<p>A You have to remove the existing runtime library from DFHRPL &amp; STEPLIB/LNKLST and use the OS/VS COBOL runtime provided within the Language Environment libraries. However, the Language Environment runtime itself is optional for CICS TS 1.3.</p>
<p>Q How do I change to using a supported OS/VS COBOL runtime for CICS TS 2.3?</p>	<p>A You have to remove the existing runtime library from DFHRPL &amp; STEPLIB/LNKLST and use the OS/VS COBOL runtime provided within the Language Environment libraries. The Language Environment runtime is required for CICS TS 2.3 and so you will already have these Language Environment runtime libraries in the DFHRPL &amp; STEPLIB/LNKLST concatenations.</p>
<p>Q So it's just a JCL change?</p>	<p>A Yes: Nothing else has to be done.</p>
<p>Q How do I find which library contains the unsupported OS/VS COBOL runtime in my JCL?</p>	<p>A The library will contain ILBO* modules. It is probably called COBLIB, COB2LIB, COB2CICS or VSCLLIB.</p>
<p>Q Specifically, how do I install the supported OS/VS COBOL runtime provided by Language Environment in a CICS TS 1.3 region?</p>	<p>A You must remove the unsupported OS/VS COBOL runtime library from the DFHRPL &amp; STEPLIB/LNKLST concatenation and insert the Language Environment runtime libraries (SCEECICS and SCEERUN).</p>
<p>Q And if I want to actually use a Language Environment runtime for modules written with other languages in a CICS TS 1.3 region...?</p>	<p>A Then you have to additionally RDO define and install the required Language Environment items (group CEE is shipped in the CEECCSD member of SCEESAMP). This will generate the required effect on modules that can use the Language Environment runtime or a native runtime (like VS COBOL II modules), as they will change to using the Language Environment runtime.</p>

Q Can I use my existing unsupported OS/VS COBOL runtime together with a Language Environment runtime?	A No.
Q Do I have to recompile my OS/VS COBOL program?	A No.
Q Do I have to relink my OS/VS COBOL program?	A No.
Q Do the Language Environment parameter settings effect the execution of an OS/VS COBOL module?	A No.
Q How do I change an OS/VS COBOL program?	A You have to use a CICS translator (along with the unsupported compiler) that supports OS/VS COBOL.
Q What is the supported CICS translator for OS/VS COBOL programs?	A The translator supplied with CICS TS 1.3 is the last translator to support OS/VS COBOL.
Q This means I have to save my CICS TS 1.3 translator in order to change an OS/VS COBOL program?	A Yes: But this translator will go out of service when CICS TS 1.3 goes out of service. So you will end up using an unsupported translator and unsupported compiler.
Q If I'm using this CICS TS 1.3 translator, can I get newer EXEC CICS commands into an OS/VS COBOL program?	A You cannot do this. The available EXEC CICS commands will be fixed at the CICS TS 1.3 level.
Q So IBM (including CICS) will support a runtime for OS/VS COBOL module but not the method of generating the module?	A Yes: That's the unusual situation.
Q What would be the advantage in converting my OS/VS COBOL source into that which can be compiled with the latest COBOL compiler?	A Conversion away from OS/VS COBOL source is difficult due to the change in COBOL language constructs: there are tools that can assist in this conversion. However, in so doing you will reap advantages in avoiding the AMODE(24) restriction for OS/VS COBOL and removing the processing overhead associated with establishing a pseudo reentrant environment for the execution of each module within the CICS TS region.

Q Can I use the integrated translator?	A No.
Q Where is the documentation for this?	<p>A SC28-1944-03: Language Environment for z/OS and OS/390 Run-Time Migration Guide.</p> <p>GC27-1409-00: Enterprise COBOL for z/OS and OS/390 Compiler and Run-Time Migration Guide.</p>
Q Will CICS TS >2.3 support running OS/VS COBOL modules?	<p>No: CICS TS 2.3 is the last release of CICS to support the running of OS/VS COBOL modules. If conversion for use with later COBOL compilers is not possible, customers may have support problems after CICS TS 2.3 becomes unsupported.</p>
Q What's the bottom line?	<p>A Just remove the old OS/VS COBOL runtime library from the DFHRPL &amp; STEPLIB/LNKLST concatenations. Use the Language Environment runtime library instead and all should be well.</p>

## Considerations for VS COBOL II programs

This section describes migration from a CICS TS region without a Language Environment runtime to the case where the Language Environment runtime is being used for modules compiled by:

### Either mode COBOL Compilers

VS COBOL II (5668-022)  
VS COBOL II (5668-023)  
VS COBOL II (5668-958)

Modules produced by these compilers can run within a VS COBOL II runtime or within the Language Environment runtime. This section assumes that they are not currently using the Language Environment runtime. When running with CICS TS 1.3, the choice of runtime is made by CICS (if the Language Environment runtime is present it will be used for these modules). When running within CICS TS 2.3, the Language Environment runtime will always be used for VS COBOL II modules.

Modules compiled with the following (later) compilers require the Language Environment runtime. This means modules compiled with the above compilers will already be using the Language Environment runtime if any modules compiled with the following are present within the CICS TS region):

### COBOL Compilers requiring the Language Environment runtime

SAA AD/Cycle COBOL/370 (5688-197)  
SAA AD/Cycle COBOL/370 (5668-958)  
COBOL for MVS & VM (5688-197)  
COBOL for OS/390 & VM V2 (5648-A25)  
COBOL for OS/390 & VM V2 (5648-A25) with APAR PQ45462  
Enterprise COBOL for z/OS & OS/390 V3 (5655-G53)

The Language Environment runtime is required for CICS TS 2.3, so VS COBOL II modules will not use their own runtime in this, and subsequent, releases.

## VS COBOL II

- |   |  |
|---|--|
| Q How does CICS execute VS COBOL II modules in CICS TS 1.3?   | A It depends upon whether or not the Language Environment runtime is present.<br>If CICS determines that the module is VS COBOL II and the Language Environment runtime is present, then the VS COBOL II module will run within the Language Environment runtime.<br>If the Language Environment runtime is not present, then the (unsupported) VS COBOL II runtime will be used for the module. |
| Q How does CICS execute VS COBOL II modules in CICS TS 2.3?   | A As the Language Environment runtime is required, the VS COBOL II module will run within the Language Environment runtime.  |
| Q Can I force one runtime to be used?   | A No: You cannot alter this aspect of CICS' processing.  |
| Q Can I run a VS COBOL II module using an unsupported VS COBOL II runtime within CICS TS 1.3 when the Language Environment libraries are not present? | A Yes: They will run without the need to recompile or relink. But you should not be using an unsupported runtime.  |
| Q Can I run a VS COBOL II module using a supported VS COBOL II runtime within CICS TS 1.3 when the Language Environment libraries are not present?    | A No: There is not a supported runtime in this case. The preferred method is to use the Language Environment runtime for VS COBOL II modules.  |
| Q Can I run a VS COBOL II module using an unsupported runtime within CICS TS 1.3 when the Language Environment runtime is present?                    | A No: You will be running within the Language Environment runtime and so not using the unsupported runtime.  |
| Q Can I run a VS COBOL II using an unsupported runtime within CICS TS 2.3?  | A No: The Language Environment runtime is required for CICS TS 2.3 and so VS COBOL II modules have to execute within the Language Environment runtime.   |

Q How do I change to using a supported runtime for CICS TS 1.3?

A You have to install the Language Environment runtime within CICS. Remove the existing VS COBOL II library from the DFHRPL & STEPLIB/LNKLST concatenations and insert the Language Environment runtime libraries (SCEERUN & SCEECICS). Install the Language Environment RDO group (group CEE is shipped in member CEECCSD of SCEESAMP).

Q Anything else?

A Yes: The Language Environment options table (CEEROPT) may need compiling to set regionwide defaults. As part of Language Environment installation the CEECOPT table contains Language Environment runtime options that control operations for all z/OS regions. The CEEROPT table overrides CEECOPT settings for all CICS TS regions.

Q And that's it?

A No: Things are more complicated than that. The VS COBOL II module runtime behaviour may change.

Q Do VS COBOL II programs run in exactly the same manner between using an unsupported runtime and using the supported Language Environment runtime?

A No: It depends upon the regionwide Language Environment settings (in CEECOPT and CEEROPT) and coding techniques. The default Language Environment options are not equivalent to those used by the VS COBOL II runtime. You will have to carefully set the regionwide runtime options.



Q Care to be less vague?

A There is one noticeable change between the VS COBOL II runtime and the Language Environment runtime (using default Language Environment settings) to do with the way obtained storage is cleared. VS COBOL II programs may assume that this is always set to X'00' whereas within the Language Environment runtime the default is not to do this clearing.

The relevant settings are `WSCLEAR` within the VS COBOL II program and the Language Environment `STORAGE` value.

Q So some VS COBOL II modules may assume cleared areas and so go wrong?

A Yes: Unfortunate things may occur.

Q Any other operational changes?

A Some other runtime things will have changed, but these should not have an effect on the execution of a VS COBOL II module as they are mostly system-related performance options. Again, the Language Environment `CEECOPT` & `CEEROPT` tables can be used to control the Language Environment runtime for CICS TS.

The most important Language Environment parameter setting in this area is `CBLPSHPOP`, which has an effect when the VS COBOL II module calls another module without using an `EXEC CICS LINK`.

The usage of virtual storage will be increased, and so CICS' `EDSALIM` setting will probably need to be increased.

SC28-1944-03: Language Environment for z/OS and OS/390 Run-Time Migration Guide covers these effects in detail.

Q	What about Abend handling within the Language Environment runtime?	A	The Abend codes will change from those used by the VS COBOL II runtime to those used by the Language Environment runtime. Consequently, VS COBOL II programs handling abends may need to change. Again, Language Environment runtime options and exits can be used to control behaviour.
Q	How do I set the Language Environment runtime options for my CICS region to be those suitable for VS COBOL II modules, bearing in mind that these will also affect other modules?	A	Code up the CEEROPT Language Environment parameter table. These Language Environment options do not apply to OS/VS COBOL modules, but will affect all modules using the Language Environment runtime.
Q	Do I have to recompile my VS COBOL II program to change to the Language Environment runtime?	A	No.
Q	Do I have to relink my VS COBOL II program to change to the Language Environment runtime?	A	No. But if you need a specific Language Environment setting, you may have to relink with a CEEUOPT table.
Q	How can I migrate from the VS COBOL II runtime to a Language Environment runtime for a specific VS COBOL II module?	A	You cannot do this as CICS selects the Language Environment runtime (in CICS TS 1.3) if it is present for all VS COBOL II modules in the CICS TS region.
Q	But I don't want to migrate all my VS COBOL II programs at once...?	A	In this case you will have to separate items into two CICS regions one having the Language Environment runtime and one without it (so using the unsupported VS COBOL II runtime).
Q	How do I change a VS COBOL II program?	A	You have to use a CICS translator (along with the unsupported compiler), which supports VS COBOL II.
Q	What is the supported CICS translator for VS COBOL II programs?	A	The translator supplied with CICS TS 2.2 is the last translator to support VS COBOL II as such.

<p>Q This means I have to save my CICS TS 2.2 (or CICS TS 1.3) translator in order to change a VS COBOL II program?</p>	<p>A No. VS COBOL II programs use the COBOL2 translator setting which is not supported in CICS TS 2.3. However, the COBOL3 translator setting can be used to compile a VS COBOL II program.</p>
<p>Q But if I change the translator setting, I'm also changing the program?</p>	<p>A Yes: But as the COBOL3 setting does not, essentially, do anything differently for programs using COBOL2 and written to the VS COBOL II standard, this is a cosmetic change. However, if you are changing the VS COBOL II program, you should be using the latest supported COBOL compiler, and so going away from a VS COBOL II situation thus generating a properly Language Environment-enabled module.</p>
<p>Q If I'm going to use an unsupported VS COBOL II compiler on source with the COBOL2 translator option, so using the CICS TS 2.2 (or CICS TS 1.3) translator, how can I get newer EXEC CICS commands into a VS COBOL II program?</p>	<p>A You cannot do this. The available EXEC CICS commands will be fixed at the CICS TS 2.2 (or CICS TS 1.3) level.</p>
<p>Q So IBM (including CICS) will support a Language Environment runtime for VS COBOL II but not the method of generating the module?</p>	<p>A Yes: that's the unusual situation. But, as the COBOL3 setting can be used with a supported compiler, everything is really supported.</p>
<p>Q Can I use the integrated translator?</p>	<p>A No.</p>
<p>Q Will CICS TS &gt;2.3 support running VS COBOL II modules?</p>	<p>Yes: They will be supported in future releases of CICS (using the Language Environment runtime).</p>
<p>Q Where is the documentation for this?</p>	<p>A SC28-1944-03: Language Environment for z/OS and OS/390 Run-Time Migration Guide.</p> <p>GC27-1409-00: Enterprise COBOL for z/OS and OS/390 Compiler and Run-Time Migration Guide.</p>

Q What's the bottom line?

A Installing the Language Environment runtime has implications for the execution of VS COBOL II modules that depend upon Language Environment parameter settings (in particular, the STORAGE option).

## Considerations for PL/I programs

This section describes migration considerations from a CICS TS region without the Language Environment runtime to the Language Environment runtime for modules compiled by:

### Either mode PL/I Compilers

- OS PL/I Optimizing Compiler Version 1 (5724-PLI)
- OS PL/I Optimizing Compiler Version 2 (5668-909)
- OS PL/I Optimizing Compiler Version 2 (5668-910)
- OS PL/I Optimizing Compiler Version 2 (5668-911)

Modules produced by these compilers can run within a PL/I runtime or within the Language Environment runtime. This section assumes that they are not currently using the Language Environment runtime. In CICS TS 1.3, the choice of runtime is made by CICS (if the Language Environment runtime is present it will be used for these modules).

Modules compiled with the following (later) compilers require the Language Environment runtime. This means modules compiled with the above compilers will already be using the Language Environment runtime if any modules compiled with the following are present within the CICS TS region):

### PL/I Compilers requiring the Language Environment runtime

- SAA AD/Cycle PL/I for MVS and VM (5688-235)
- PL/I for MVS & VM Version 1 (5688-235)
- Visual Age PL/I for OS/390 Version 2 (5655-B22)
- Enterprise PL/I for z/OS & OS/390 Version 3 (5655-H31)

The Language Environment runtime is required for CICS TS 2.3, so PL/I modules will not use their own runtime in this, and subsequent, releases.

## OS PL/I

- |   |  |   |   |
|---|--|---|---|
| Q | How does CICS execute OS PL/I modules in CICS TS 1.3?  | A | It depends upon whether or not the Language Environment runtime is present.<br>If CICS determines that the module is OS PL/I and the Language Environment runtime is present, then the OS PL/I module will run within the Language Environment runtime.<br>If the Language Environment runtime is not present, the (unsupported) OS PL/I runtime will be used for the module. |
| Q | Can I force one runtime to be used?  | A | No: You cannot alter this aspect of CICS' processing.   |
| Q | Can I run an OS PL/I module using an unsupported runtime within CICS TS 1.3 when the Language Environment libraries are not present? | A | Yes: They will run without the need to recompile or relink. But you should not be using an unsupported runtime.   |
| Q | Can I run an OS PL/I module using a supported runtime within CICS TS 1.3 when the Language Environment libraries are not present?    | A | No: There is not a supported runtime in this case. The preferred method is to use the Language Environment runtime for OS PL/I modules.   |
| Q | Can I run an OS PL/I module using an unsupported runtime within CICS TS 1.3 when the Language Environment runtime is present?        | A | No: You will be running within the Language Environment runtime and so not using the unsupported runtime.   |
| Q | Can I run an OS PL/I module using an unsupported runtime within CICS TS 2.3?   | A | No: The Language Environment runtime is required for CICS TS 2.3 and so OS PL/I modules have to run within the Language Environment runtime.  |

Q How do I change to using a supported runtime for CICS TS 1.3?

A You have to install the Language Environment runtime within CICS. Remove the existing OS PL/I library from the DFHRPL & STEPLIB/LNKLST concatenations and insert the Language Environment runtime libraries (SCEERUN & SCEEICICS). Install the Language Environment RDO group (group CEE is shipped in member CEECCSD of SCEESAMP).

Q Anything else?

A Yes: The Language Environment options table (CEEROPT) may need compiling to set regionwide defaults. As part of Language Environment installation the CEECOPT table contains Language Environment runtime options that control operations for all z/OS regions. The CEEROPT table overrides CEECOPT settings for all CICS TS regions.

Q And that's it?

A No: Things are more complicated than that. The OS PL/I runtime behaviour may change. However, there is less of an impact upon OS PL/I programs than for VS COBOL II programs as OS PL/I is stricter in this respect.

Q Do OS PL/I programs run in exactly the same manner between using an unsupported runtime and within the supported Language Environment runtime?

A No: It depends upon the regionwide Language Environment settings (in CEECOPT and CEEROPT) and coding techniques. However, they are mostly equivalent and so this should not be a problem.

The default Language Environment options are not equivalent to those used by the VS COBOL II runtime so Language Environment runtime options set for VS COBOL II may not be suitable for OS PL/I. You will have to carefully set the regionwide runtime options.

<p>Q Any operational changes?</p>	<p>A Some other runtime things will have changed, but these should not have an effect on the execution of the OS PL/I module as they are mostly system-related performance options. The Language Environment runtime options and exits can be used to control the Language Environment runtime.</p>
<p>Q What about Abend handling within the Language Environment runtime?</p>	<p>A The Abend codes will change from those used by the OS PL/I runtime to those used by the Language Environment runtime. Consequently, OS PL/I programs handling abends may need to change. Again, Language Environment runtime options and exits can be used to control things.</p>
<p>Q How do I set the Language Environment runtime options for my CICS region to be those suitable for OS PL/I modules, bearing in mind that these will also affect other modules?</p>	<p>A Code up the CEEROPT Language Environment parameter table. These Language Environment options do not apply to OS/VS COBOL modules, but will to all other modules using the Language Environment runtime.</p>
<p>Q Do I have to recompile my OS PL/I program to change to the Language Environment runtime?</p>	<p>A No. However, if you need a specific Language Environment parameter setting for the OS PL/I module you can set this in the PLIXOPT external variable.</p>
<p>Q Do I have to relink my OS PL/I program to change to the Language Environment?</p>	<p>A No. But if you need a specific Language Environment setting, you may have to relink with a CEEUOPT table.</p>
<p>Q How can I migrate from the OS PL/I runtime to a Language Environment runtime for a specific OS PL/I module?</p>	<p>A You cannot do this as CICS selects the Language Environment runtime (if it is present in CICS TS 1.3) for all OS PL/I modules in the CICS TS region.</p>
<p>Q But I don't want to migrate all my OS PL/I programs at once...?</p>	<p>A In this case you will have to separate items into two CICS regions one having the Language Environment runtime and one without it (so using the unsupported OS PL/I runtime).</p>



Q	How do I change an OS PL/I program?	A	You have to use a CICS translator (along with the unsupported compiler) which supports OS PL/I.
Q	What is the supported CICS translator for OS PL/I programs?	A	All CICS translators will support the use of PL/I. It does not matter if the compiler is supported or not.
Q	What is the implication of changing from an unsupported OS PL/I compiler to the latest supported PL/I compiler?	A	None. The latest PL/I compiler should work on older source without any problem. Using the latest compiler turns the generated module into a properly Language Environment-enabled item.
Q	So IBM (including CICS) will support the runtime and translator for OS PL/I but not the Compiler?	A	Yes: That's the unusual situation. But why should you want to use an unsupported compiler when the supported ones produce the same thing?
Q	Can I use the integrated translator?	A	No.
Q	Will CICS TS >2.3 support running OS PL/I modules?	A	No: They will not be supported in future releases of CICS. Customers should recompile with the latest PL/I compiler.
Q	Where is the documentation for this?	A	SC28-1944-03: Language Environment for z/OS and OS/390 Run-Time Migration Guide.  GC27-1458-00: Enterprise PL/I for z/OS and OS/390 Compiler and Run-Time Migration Guide.
Q	What's the bottom line?	A	OS/PLI modules should run without any problems within the Language Environment runtime.

## Considerations for C/370 programs

This section describes migration considerations from a CICS TS region without the Language Environment runtime to the Language Environment runtime for modules compiled by:

### Either mode C Compilers

C/370 Version 1 (5688-040)  
C/370 Version 2 (5688-187)  
C/370 Version 2 (5688-188)

Modules produced by these early C compilers can run within a C runtime or within the Language Environment runtime. This section assumes that they are not currently using the Language Environment runtime. Within CICS TS 1.3, the choice of runtime is made by CICS (if the Language Environment runtime is present it will be used for these modules).

Modules (including all C++) compiled with the following (later) compilers require the Language Environment runtime. This means modules compiled with the above compilers will already be using the Language Environment runtime if any modules compiled with the following are present within the CICS TS region):

### C Compilers requiring the Language Environment runtime

SAA AD/Cycle C/370 (5688-216)  
C/C++ for MVS/ESA (5655-121)  
C/C++ for OS/390 (component of 5647-A01)  
C/C++ for z/OS and OS/390 (component of 5694-A01)

The Language Environment runtime is required for CICS TS 2.3, so C/C++ modules will not use their own runtime in this, and subsequent, releases.

## OS C

- |  |   |
|--|---|
| <p>Q How does CICS execute OS C modules in CICS TS 1.3?</p>  | <p>A It depends upon whether or not the Language Environment runtime is present.<br/>If CICS determines that the module is OS C and the Language Environment runtime is present, then the OS C module will run within the Language Environment runtime.<br/>If the Language Environment runtime is not present, the OS C runtime will be used for the module.</p> |
| <p>Q Can I force one runtime to be used?</p>   | <p>A No: You cannot alter this aspect of CICS' processing.</p>  |
| <p>Q Can I run an OS C module using an unsupported runtime within CICS TS 1.3 when the Language Environment libraries are not present?</p> | <p>A Yes: They will run without the need to recompile or relink. But you should not be using an unsupported runtime.</p>  |
| <p>Q Can I run an OS C module using a supported runtime within CICS TS 1.3 when the Language Environment libraries are not present?</p>    | <p>A No: There is not a supported runtime in this case. The preferred method is to use the Language Environment runtime for OS C modules.</p>   |
| <p>Q Can I run an OS C module using an unsupported runtime within CICS TS 1.3 when the Language Environment runtime is present?</p>        | <p>A No: You will be running within the Language Environment runtime and so not using the unsupported runtime.</p>  |
| <p>Q Can I run an OS C module using an unsupported runtime within CICS TS 2.3?</p>   | <p>A No: The Language Environment runtime is required for CICS TS 2.3 and so OS C modules have to run within the Language Environment runtime.</p>  |

Q How do I change to using a supported runtime for CICS TS 1.3?

A You have to install the Language Environment runtime within CICS. Remove the existing OS C library from the DFHRPL & STEPLIB/LNKLST concatenations and insert the Language Environment runtime libraries (SCEERUN & SCEEICICS). Install the Language Environment RDO group (group CEE is shipped in member CEECCSD of SCEESAMP).

Q Anything else?

A Yes: The Language Environment options table (CEEROPT) may need compiling to set regionwide defaults. As part of Language Environment installation the CEECOPT table contains Language Environment runtime options that control operations for all z/OS regions. The CEEROPT table overrides CEECOPT settings for all CICS TS regions.

Q And that's it?

A No: Things are more complicated than that. The OS C runtime behaviour may change. However, there should be much less of an impact upon OS C programs than for VS COBOL II programs as OS C is stricter in this respect and the compilers were designed with the Language Environment runtime in mind.

<p>Q Do OS C programs run in exactly the same manner between using an unsupported runtime and within the supported Language Environment runtime?</p>	<p>A No: It depends upon the region wide Language Environment settings (in CEECOPT and CEEROPT) and coding techniques. However, they are mostly equivalent and so this should not be a problem.</p> <p>The default Language Environment options are not equivalent to those used by the VS COBOL II runtime so Language Environment runtime options set for VS COBOL II may not be suitable for OS C. You will have to carefully set the regionwide runtime options.</p>
<p>Q Any operational changes?</p>	<p>A Some other runtime things will have changed, but these should not have an effect on the execution of the OS C module, as they are mostly system-related performance options. The Language Environment runtime options can be used to control the Language Environment runtime.</p>
<p>Q How do I set the Language Environment runtime options for my CICS region to be those suitable for OS C modules, bearing in mind that these will also affect other modules?</p>	<p>A Code up the CEEROPT Language Environment parameter table. These Language Environment options do not apply to OS/VS COBOL modules, but will to all other modules using the Language Environment runtime.</p>
<p>Q Do I have to recompile my OS C program to change to the Language Environment runtime?</p>	<p>A No.</p>
<p>Q Do I have to relink my OS C program to change to the Language Environment?</p>	<p>A No. But if you need a specific Language Environment setting, you may have to relink with a CEEUOPT table.</p>
<p>Q How can I migrate from the OS C runtime to a Language Environment runtime for a specific OS C module?</p>	<p>A You cannot do this as CICS selects the Language Environment runtime if it is present for all OS C modules in the CICS TS region.</p>

Q	But I don't want to migrate all my OS C programs at once...?	A	In this case you will have to separate items into two CICS regions one having the Language Environment runtime and one without it (so using the unsupported OS C runtime).
Q	How do I change an OS C program?	A	You have to use a CICS translator (along with the unsupported compiler) which supports C.
Q	What is the supported CICS translator for OS C programs?	A	All CICS translators will support the use of C. It does not matter if the compiler is supported or not.
Q	What is the implication of changing from an unsupported OS C compiler to the latest supported C compiler?	A	None. The latest C (or C++) compiler should work on older source without any problem. Using the latest compiler turns the generated module into a properly Language Environment-enabled item.
Q	So IBM (including CICS) will support the runtime and translator for OS C but not the Compiler?	A	Yes: That's the unusual situation. But why should you want to use an unsupported compiler when the supported ones produce the same thing?
Q	Can I use the integrated translator?	A	No.
Q	Will CICS TS >2.3 support running OS C modules?	A	No: They will not be supported in future releases of CICS. Customers should recompile modules with the latest C/C++ compiler.
Q	Where is the documentation for this?	A	SC28-1944-03: Language Environment for z/OS and OS/390 Run-Time Migration Guide.  GC09-4913-00: C/C++ Compiler for z/OS and OS/390 Compiler and Run-Time Migration Guide.
Q	What's the bottom line?	A	OS C modules should run without any problems within the Language Environment runtime.

## CICS Translator support

This section describes support for high-level languages from the view of the CICS Translator. It shows the levels of CICS TS translators that can be used to compile COBOL, PL/I, C or C++ application programs.

Compiler	In Service?	CICS Translator that can be used for compilation		
		1.3	2.2	2.3
OS/VS COBOL (5740-CB1)	✗	Y1	N	N
OS/VS COBOL (5734-CB4)	✗	Y1	N	N
OS/VS COBOL (5740-LM1)	✗	Y1	N	N
VS COBOL II (5668-022)	✗	Y2	Y3	Y3
VS COBOL II (5668-023)	✗	Y2	Y3	Y3
VS COBOL II (5668-958)	✗	Y2	Y3	Y3
SAA AD/Cycle COBOL/370 (5688-197)	✗	Y2	Y3	Y3
SAA AD/Cycle COBOL/370 (5668-958)	✗	Y2	Y3	Y3
COBOL for MVS & VM (5688-197)	✗	Y2	Y3	Y3
COBOL for OS/390 & VM V2 (5648-A25)	✓	Y2	Y3	Y3
COBOL for OS/390 & VM V2 (5648-A25) with APAR PQ45462	✓	Y2	Y3	Y3
Enterprise COBOL for z/OS & OS/390 V3 (5655-G53)	✓	Y3	Y3	Y3
OS PL/I Optimizing Compiler Version 1 (5724-PLI)	✗	Y	Y	Y
OS PL/I Optimizing Compiler Version 2 (5668-909)	✗	Y	Y	Y
OS PL/I Optimizing Compiler Version 2 (5668-910)	✗	Y	Y	Y
OS PL/I Optimizing Compiler Version 2 (5668-911)	✗	Y	Y	Y
SAA AD/Cycle PL/I for MVS and VM (5688-235)	✗	Y	Y	Y
PL/I for MVS & VM Version 1 (5688-235)	✗	Y	Y	Y
Visual Age PL/I for OS/390 Version 2 (5655-B22)	✗	Y	Y	Y
Enterprise PL/I for z/OS & OS/390 Version 3 (5655-H31)	✓	Y	Y	Y
C/370 Version 1 (5688-040)	✗	Y	Y	Y
C/370 Version 2 (5688-187)	✗	Y	Y	Y
C/370 Version 2 (5688-188)	✗	Y	Y	Y
SAA AD/Cycle C/370 (5688-216)	✗	Y	Y	Y
C/C++ for MVS/ESA (5655-121)	✗	Y	Y	Y
C/C++ for OS/390 (component of 5647-A01)	✓	Y	Y	Y
C/C++ for z/OS and OS/390 (component of 5694-A01)	✓	Y	Y	Y

Y = Yes; N = No; Y1 = Yes, with the COBOL option; Y2=Yes, with the COBOL2 option; Y3=Yes, with the COBOL3 option.

## CICS runtime support for unsupported native libraries

This section describes support for execution of high-level language modules within CICS where native (non-Language Environment) libraries are employed. These native language-dependant libraries are all out of service and unsupported by IBM.

Compiler	In Service?	Usage of unsupported native runtime libraries		
		1.3	2.3	>2.3
OS/VS COBOL (5740-CB1)	✗	Y	Y@	N!
OS/VS COBOL (5734-CB4)	✗	Y	Y@	N!
OS/VS COBOL (5740-LM1)	✗	Y	Y@	N!
VS COBOL II (5668-022)	✗	Y?	N	N
VS COBOL II (5668-023)	✗	Y?	N	N
VS COBOL II (5668-958)	✗	Y?	N	N
SAA AD/Cycle COBOL/370 (5688-197)	✗	N	N	N
SAA AD/Cycle COBOL/370 (5668-958)	✗	N	N	N
COBOL for MVS & VM (5688-197)	✗	N	N	N
COBOL for OS/390 & VM V2 (5648-A25)	✓	N	N	N
COBOL for OS/390 & VM V2 (5648-A25) with APAR PQ45462	✓	N	N	N
Enterprise COBOL for z/OS & OS/390 V3 (5655-G53)	✓	N	N	N
OS PL/I Optimizing Compiler Version 1 (5724-PLI)	✗	Y?	N	N!
OS PL/I Optimizing Compiler Version 2 (5668-909)	✗	Y?	N	N!
OS PL/I Optimizing Compiler Version 2 (5668-910)	✗	Y?	N	N!
OS PL/I Optimizing Compiler Version 2 (5668-911)	✗	Y?	N	N!
SAA AD/Cycle PL/I for MVS and VM (5688-235)	✗	N	N	N
PL/I for MVS & VM Version 1 (5688-235)	✗	N	N	N
Visual Age PL/I for OS/390 Version 2 (5655-B22)	✗	N	N	N
Enterprise PL/I for z/OS & OS/390 Version 3 (5655-H31)	✓	N	N	N
C/370 Version 1 (5688-040)	✗	Y?	N	N!
C/370 Version 2 (5688-187)	✗	Y?	N	N!
C/370 Version 2 (5688-188)	✗	Y?	N	N!
SAA AD/Cycle C/370 (5688-216)	✗	N	N	N
C/C++ for MVS/ESA (5655-121)	✗	N	N	N
C/C++ for OS/390 (component of 5647-A01)	✓	N	N	N
C/C++ for z/OS and OS/390 (component of 5694-A01)	✓	N	N	N

Y = Yes, but not recommended; N = No (an unsupported native runtime cannot be used); Y@ = Yes, but requires special consideration and not recommended; Y? = Yes, provided the Language Environment runtime is absent and not recommended; N! = No, as support will be withdrawn in the next release of CICS after CICS TS 2.3.

Implications for CICS TS modules when Language Environment is installed

CICS runtime support for unsupported native libraries



## CICS runtime support for supported native libraries

This section describes support for execution of high-level language modules within CICS where a native runtime provided within the supported Language Environment libraries is employed.

Compiler	Usage of supported native runtimes from within Language Environment libraries				
		In Service?	1.3	2.3	>2.3
OS/VS COBOL (5740-CB1)		✗	Y	Y	N!
OS/VS COBOL (5734-CB4)		✗	Y	Y	N!
OS/VS COBOL (5740-LM1)		✗	Y	Y	N!
VS COBOL II (5668-022)		✗	Y?	N	N
VS COBOL II (5668-023)		✗	Y?	N	N
VS COBOL II (5668-958)		✗	Y?	N	N
SAA AD/Cycle COBOL/370 (5688-197)		✗	N	N	N
SAA AD/Cycle COBOL/370 (5668-958)		✗	N	N	N
COBOL for MVS & VM (5688-197)		✗	N	N	N
COBOL for OS/390 & VM V2 (5648-A25)		✓	N	N	N
COBOL for OS/390 & VM V2 (5648-A25) with APAR PQ45462		✓	N	N	N
Enterprise COBOL for z/OS & OS/390 V3 (5655-G53)		✓	N	N	N
OS PL/I Optimizing Compiler Version 1 (5724-PLI)		✗	Y?	N	N!
OS PL/I Optimizing Compiler Version 2 (5668-909)		✗	Y?	N	N!
OS PL/I Optimizing Compiler Version 2 (5668-910)		✗	Y?	N	N!
OS PL/I Optimizing Compiler Version 2 (5668-911)		✗	Y?	N	N!
SAA AD/Cycle PL/I for MVS and VM (5688-235)		✗	N	N	N
PL/I for MVS & VM Version 1 (5688-235)		✗	N	N	N
Visual Age PL/I for OS/390 Version 2 (5655-B22)		✗	N	N	N
Enterprise PL/I for z/OS & OS/390 Version 3 (5655-H31)		✓	N	N	N
C/370 Version 1 (5688-040)		✗	Y?	N	N!
C/370 Version 2 (5688-187)		✗	Y?	N	N!
C/370 Version 2 (5688-188)		✗	Y?	N	N!
SAA AD/Cycle C/370 (5688-216)		✗	N	N	N
C/C++ for MVS/ESA (5655-121)		✗	N	N	N
C/C++ for OS/390 (component of 5647-A01)		✓	N	N	N
C/C++ for z/OS and OS/390 (component of 5694-A01)		✓	N	N	N

Y = Yes; N = No (a Language Environment runtime must be used); Y? = Yes, provided the Language Environment runtime is absent; N! = No, as support will be withdrawn in the next release of CICS after CICS TS 2.3.

Implications for CICS TS modules when Language Environment is installed

CICS runtime support for supported native libraries

## CICS support for the Language Environment runtime

This section shows support for execution of high-level language modules within CICS where they are running within the Language Environment runtime.

Compiler	In Service?	Usage of Language Environment runtime		
		1.3	2.3	>2.3
OS/VS COBOL (5740-CB1)	✗	N	N	N!
OS/VS COBOL (5734-CB4)	✗	N	N	N!
OS/VS COBOL (5740-LM1)	✗	N	N	N!
VS COBOL II (5668-022)	✗	Y?	Y	Y
VS COBOL II (5668-023)	✗	Y?	Y	Y
VS COBOL II (5668-958)	✗	Y?	Y	Y
SAA AD/Cycle COBOL/370 (5688-197)	✗	Y	Y	Y
SAA AD/Cycle COBOL/370 (5668-958)	✗	Y	Y	Y
COBOL for MVS & VM (5688-197)	✗	Y	Y	Y
COBOL for OS/390 & VM V2 (5648-A25)	✓	Y	Y	Y
COBOL for OS/390 & VM V2 (5648-A25) with APAR PQ45462	✓	Y	Y	Y
Enterprise COBOL for z/OS & OS/390 V3 (5655-G53)	✓	Y	Y	Y
OS PL/I Optimizing Compiler Version 1 (5724-PLI)	✗	Y?	Y	N!
OS PL/I Optimizing Compiler Version 2 (5668-909)	✗	Y?	Y	N!
OS PL/I Optimizing Compiler Version 2 (5668-910)	✗	Y?	Y	N!
OS PL/I Optimizing Compiler Version 2 (5668-911)	✗	Y?	Y	N!
SAA AD/Cycle PL/I for MVS and VM (5688-235)	✗	Y	Y	Y
PL/I for MVS & VM Version 1 (5688-235)	✗	Y	Y	Y
Visual Age PL/I for OS/390 Version 2 (5655-B22)	✗	Y	Y	Y
Enterprise PL/I for z/OS & OS/390 Version 3 (5655-H31)	✓	Y	Y	Y
C/370 Version 1 (5688-040)	✗	Y?	Y	N!
C/370 Version 2 (5688-187)	✗	Y?	Y	N!
C/370 Version 2 (5688-188)	✗	Y?	Y	N!
SAA AD/Cycle C/370 (5688-216)	✗	Y	Y	Y
C/C++ for MVS/ESA (5655-121)	✗	Y	Y	Y
C/C++ for OS/390 (component of 5647-A01)	✓	Y	Y	Y
C/C++ for z/OS and OS/390 (component of 5694-A01)	✓	Y	Y	Y

Y = Yes; N = No; Y? = Yes, provided the CEE RDO group is active; N! = No, as support will be withdrawn in the next release of CICS after CICS TS 2.3.

Implications for CICS TS modules when Language Environment is installed

CICS support for the Language Environment runtime

## CICS Integrated Translator support

This section shows which compilers can use the CICS TS 2.3 Integrated Translator.

Compiler			Can use the CICS TS 2.3 Integrated Translator?
	In Service?		
OS/VS COBOL (5740-CB1)	✗		N
OS/VS COBOL (5734-CB4)	✗		N
OS/VS COBOL (5740-LM1)	✗		N
VS COBOL II (5668-022)	✗		N
VS COBOL II (5668-023)	✗		N
VS COBOL II (5668-958)	✗		N
SAA AD/Cycle COBOL/370 (5688-197)	✗		N
SAA AD/Cycle COBOL/370 (5668-958)	✗		N
COBOL for MVS & VM (5688-197)	✗		N
COBOL for OS/390 & VM V2 (5648-A25)	✓		N
COBOL for OS/390 & VM V2 (5648-A25) with APAR PQ45462	✓		Y
Enterprise COBOL for z/OS & OS/390 V3 (5655-G53)	✓		Y
OS PL/I Optimizing Compiler Version 1 (5724-PLI)	✗		N
OS PL/I Optimizing Compiler Version 2 (5668-909)	✗		N
OS PL/I Optimizing Compiler Version 2 (5668-910)	✗		N
OS PL/I Optimizing Compiler Version 2 (5668-911)	✗		N
SAA AD/Cycle PL/I for MVS and VM (5688-235)	✗		N
PL/I for MVS & VM Version 1 (5688-235)	✗		N
Visual Age PL/I for OS/390 Version 2 (5655-B22)	✗		N
Enterprise PL/I for z/OS & OS/390 Version 3 (5655-H31)	✓		Y
C/370 Version 1 (5688-040)	✗		N
C/370 Version 2 (5688-187)	✗		N
C/370 Version 2 (5688-188)	✗		N
SAA AD/Cycle C/370 (5688-216)	✗		N
C/C++ for MVS/ESA (5655-121)	✗		N
C/C++ for OS/390 (component of 5647-A01)	✓		N
C/C++ for z/OS and OS/390 (component of 5694-A01)	✓		N

Y = Yes; N = No.

## Use of the IBM Distributed Debugger

This section shows which compiled modules can be debugged with the IBM Distributed Debugger in CICS TS 2.3.

Compiler	Can use the IBM Distributed Debugger in CICS TS 2.3?	
	In Service?	
OS/VS COBOL (5740-CB1)	✘	N
OS/VS COBOL (5734-CB4)	✘	N
OS/VS COBOL (5740-LM1)	✘	N
VS COBOL II (5668-022)	✘	Y?
VS COBOL II (5668-023)	✘	Y?
VS COBOL II (5668-958)	✘	Y?
SAA AD/Cycle COBOL/370 (5688-197)	✘	Y?
SAA AD/Cycle COBOL/370 (5668-958)	✘	Y?
COBOL for MVS & VM (5688-197)	✘	Y?
COBOL for OS/390 & VM V2 (5648-A25)	✔	Y?
COBOL for OS/390 & VM V2 (5648-A25) with APAR PQ45462	✔	Y?
Enterprise COBOL for z/OS & OS/390 V3 (5655-G53)	✔	Y
OS PL/I Optimizing Compiler Version 1 (5724-PLI)	✘	Y?
OS PL/I Optimizing Compiler Version 2 (5668-909)	✘	Y?
OS PL/I Optimizing Compiler Version 2 (5668-910)	✘	Y?
OS PL/I Optimizing Compiler Version 2 (5668-911)	✘	Y?
SAA AD/Cycle PL/I for MVS and VM (5688-235)	✘	Y?
PL/I for MVS & VM Version 1 (5688-235)	✘	Y?
Visual Age PL/I for OS/390 Version 2 (5655-B22)	✘	Y?
Enterprise PL/I for z/OS & OS/390 Version 3 (5655-H31)	✔	Y
C/370 Version 1 (5688-040)	✘	Y?
C/370 Version 2 (5688-187)	✘	Y?
C/370 Version 2 (5688-188)	✘	Y?
SAA AD/Cycle C/370 (5688-216)	✘	Y?
C/C++ for MVS/ESA (5655-121)	✘	Y?
C/C++ for OS/390 (component of 5647-A01)	✔	Y?
C/C++ for z/OS and OS/390 (component of 5694-A01)	✔	Y

Y = Yes; Y? = Yes, with restrictions; N = No.

## Use with WebSphere Studio Enterprise Developer

This section shows which modules can be created within the ambit of WebSphere Studio Enterprise Developer for use in CICS TS 2.3.

Compiler	Can be coded with WSED for CICS TS 2.3?	
	In Service?	
OS/VS COBOL (5740-CB1)	✗	N
OS/VS COBOL (5734-CB4)	✗	N
OS/VS COBOL (5740-LM1)	✗	N
VS COBOL II (5668-022)	✗	N
VS COBOL II (5668-023)	✗	N
VS COBOL II (5668-958)	✗	N
SAA AD/Cycle COBOL/370 (5688-197)	✗	N
SAA AD/Cycle COBOL/370 (5668-958)	✗	N
COBOL for MVS & VM (5688-197)	✗	N
COBOL for OS/390 & VM V2 (5648-A25)	✓	N
COBOL for OS/390 & VM V2 (5648-A25) with APAR PQ45462	✓	Y?
Enterprise COBOL for z/OS & OS/390 V3 (5655-G53)	✓	Y
OS PL/I Optimizing Compiler Version 1 (5724-PLI)	✗	N
OS PL/I Optimizing Compiler Version 2 (5668-909)	✗	N
OS PL/I Optimizing Compiler Version 2 (5668-910)	✗	N
OS PL/I Optimizing Compiler Version 2 (5668-911)	✗	N
SAA AD/Cycle PL/I for MVS and VM (5688-235)	✗	N
PL/I for MVS & VM Version 1 (5688-235)	✗	N
Visual Age PL/I for OS/390 Version 2 (5655-B22)	✗	N
Enterprise PL/I for z/OS & OS/390 Version 3 (5655-H31)	✓	Y
C/370 Version 1 (5688-040)	✗	N
C/370 Version 2 (5688-187)	✗	N
C/370 Version 2 (5688-188)	✗	N
SAA AD/Cycle C/370 (5688-216)	✗	N
C/C++ for MVS/ESA (5655-121)	✗	N
C/C++ for OS/390 (component of 5647-A01)	✓	N
C/C++ for z/OS and OS/390 (component of 5694-A01)	✓	N

Y = Yes; Y? = Yes, with restrictions; N = No.

<<<<<<<<<<<<<<<end of document>>>>>>>>>>>>>