MQSeries® Integrator Agent for CICS Transaction Server®

Product overview

Version 1 Release 1

MQSeries® Integrator Agent for CICS Transaction Server®

Product overview

Version 1 Release 1

Note: Before using this information and the product it supports, read the information in "Notices" on page 11.

Contents

Figures

Chapter 1. About MQSeries Integrator Agent for CICS Transaction Server

IBM® MQSeries Integrator Agent for CICS Transaction Server, known here also as MQSI Agent for CICS, consists of the MQSI Agent for CICS Adapter Builder and the MQSI Agent for CICS server run time.

These two components work together in a process that defines, models and executes adapters for the purpose of performing business transaction processing.

The MQSeries Integrator Agent for CICS Adapter Builder

The MQSeries Integrator Agent for CICS Adapter Builder runs on the Windows NT 4.0 platform. It has an intuitive visual interface that enables one to model an adapter to implement a business transaction process. See Figure 1 on page 2.

The MQSeries Integrator Agent for CICS Adapter Builder consists of several major components:

- *Importers* that enable a user to import an application's interface into the MQSeries Integrator Agent for CICS Adapter Builder in the form of messages, where then a user can work on it. The importers import:
  - 3270 screens from legacy CICS/IMS applications
  - COBOL record descriptions from existing CICS transactions.
- The *Control Center*, the user interface. It is similar to MQSeries Integrator's user interface. Via the Control Center, you can:
  - Control the import of messages by the importers.
  - Modify imported messages or manually create new messages.
  - Model the microflow. The model can contain a wide variety of functionalities, such as control flow, data flow, sequential navigation, conditional branching including decision and iteration, data typing, storing data context, transformation of data elements, logical operations and custom code.
  - Store your work in the repository and perform some validation.
  - Control the generation of adapter code by the generator.
- The *repository*, a database which contains the messages and adapter model.
- The generator facility that enables the adapter to behave as specified in the modeled microflow. The generator:
  - Reads definitions of the microflows and the three types of Server Adapter programs (FEPI, DPL, and MQ) from the XML documents in the repository.
  - From the definitions, generates source code in the appropriate language. This version will provide support in COBOL (source and copybooks) and job control language (JCL). This generated output is the adapter.
  - Deploys the adapter (source code, JCL and copybooks) to a specified location on the OS/390 server system.
  - Optionally, initiates the compilation of the source code into the executables of the *Navigators* and the *Server Adapter programs*.

    Note: You do not have to initiate the compile from the builder tool. You have the option to run the compile JCL from outside the build time environment.

MQSeries Integrator Agent for CICS
Transaction Server Adapter Builder

control center
• Import modify and create messages.
• Compose adapter model.
• Control import and generation.

COBOL Record Descriptions

structured data types

3270 Screens

FEPI commands

importers

COBOL

3270 Screen

repository

generator

COBOL copybooks

COBOL sources

JCL
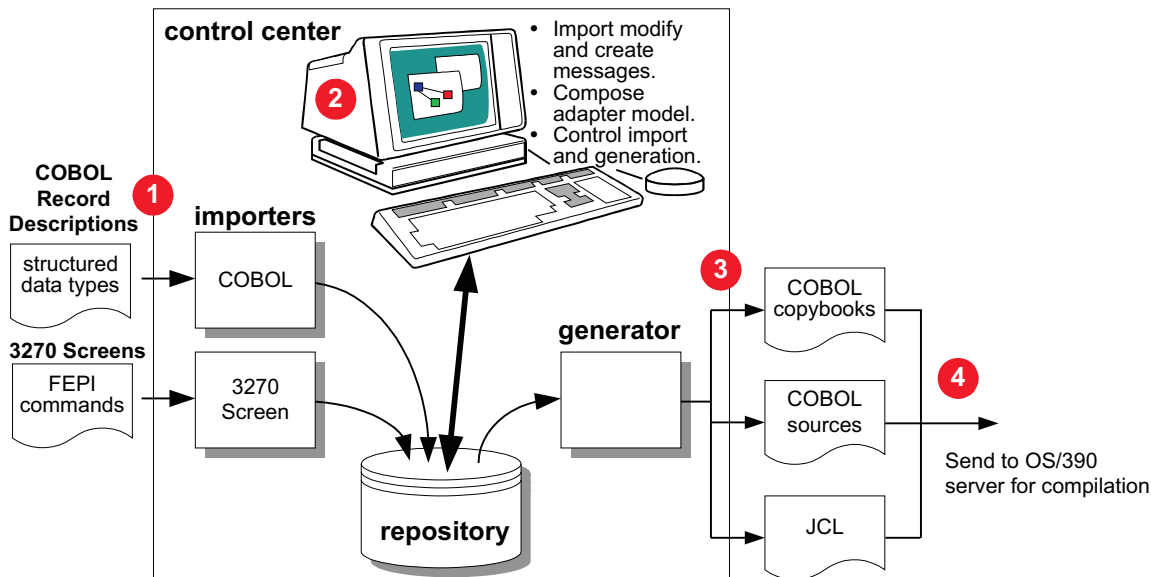
Send to OS/390 server for compilation

Figure 1. Components of the MQSeries Integrator Agent for CICS Adapter Builder

Using the builder

At build time, a person uses the MQSeries Integrator Agent for CICS Adapter Builder to perform the following tasks:

- Construct a *microflow*, the purpose of which is to model all or part of the behavior required to execute a business transaction.
- Generate COBOL source code. This source code contains the microflow as well as static server run time information, that together make up the *adapter*.
- Deploy the adapter to a specified location the OS/390 server.

A person can reuse the adapters that they have created.

Benefits of using the builder

The MQSeries Integrator Agent for CICS Adapter Builder provides several major benefits:

- It allows the user to concentrate on the application domain rather than on low level programming.
- It provides a graphical display of the adapter during build time to show the processing flow within the adapter.
- Adapters can be easily created, modified and regenerated.
- Adapter maintenance is simplified.
- The generated adapter is compiled, which provides a higher level of performance than an interpreted run time.
- Because of the builder's structured approach, the user can use the same methodology to build a variety of different adapter types.
- Because the builder's user interface is based on MQSeries Integrator, version 2.0, if you know how to use MQSeries Integrator's Control Center, using the MQSeries Integrator Agent for CICS Adapter Builder is relatively easy to learn.

MQSeries Integrator Agent for CICS at run time

Run time is when the deployed adapter becomes operational.

At run time, the adapter, performs as a BTS application, fulfilling the business transaction processes modelled at build time.

All the rules and processing associated with completing the business transaction are modelled at build time and then executed at run time.

MQSeries Integrator Agent for CICS run time processing occurs when the MQSeries Integrator Agent for CICS DPL Stub program receives a request message that originates from a controlling application.

The DPL Stub program uses information contained in the request message to define and run the BTS process, update the MQSeries Integrator Agent for CICS Properties file and to initiate the programmatic functions that will enable the adapter to process the request. The programmatic functions performed by the adapter are referred to as *adapter request processing*.

With MQSeries Integrator Agent for CICS, the adapter functionality modeled by the builder enables any MQSeries-enabled application or any application that is capable of initiating a CICS program to access:

- Existing CICS transactions via a Distributed Program Link (DPL).
- Legacy CICS/IMS applications via a 3270 data stream.
- MQSeries-enabled applications via the MQSeries product.

Several examples of adapters that implement business transactions are:

- Add a sales order.
- Check account balance.
- Update customer record.

Figure 2. Adapter modeling and run time execution.
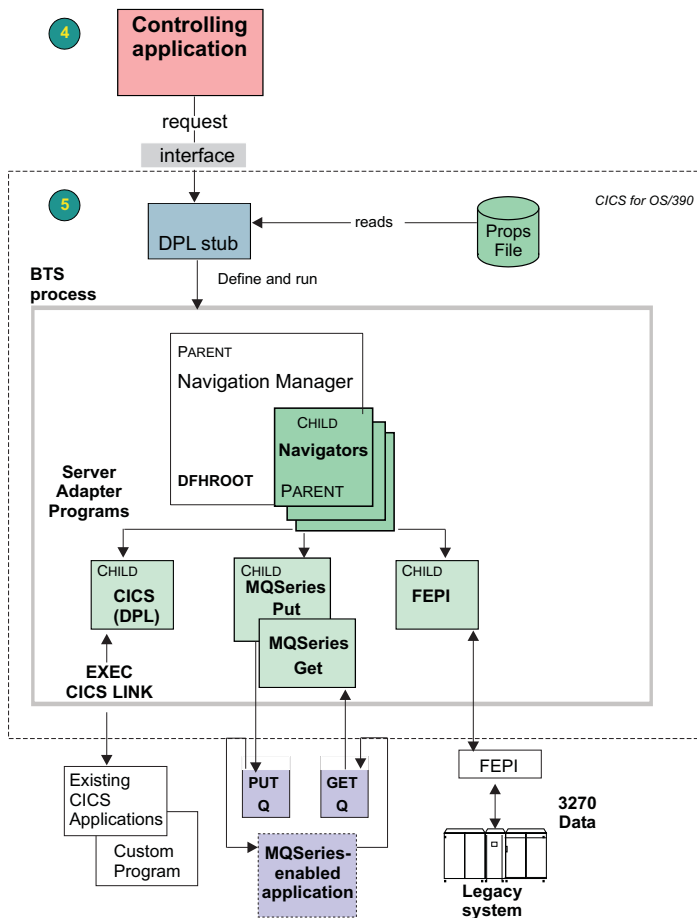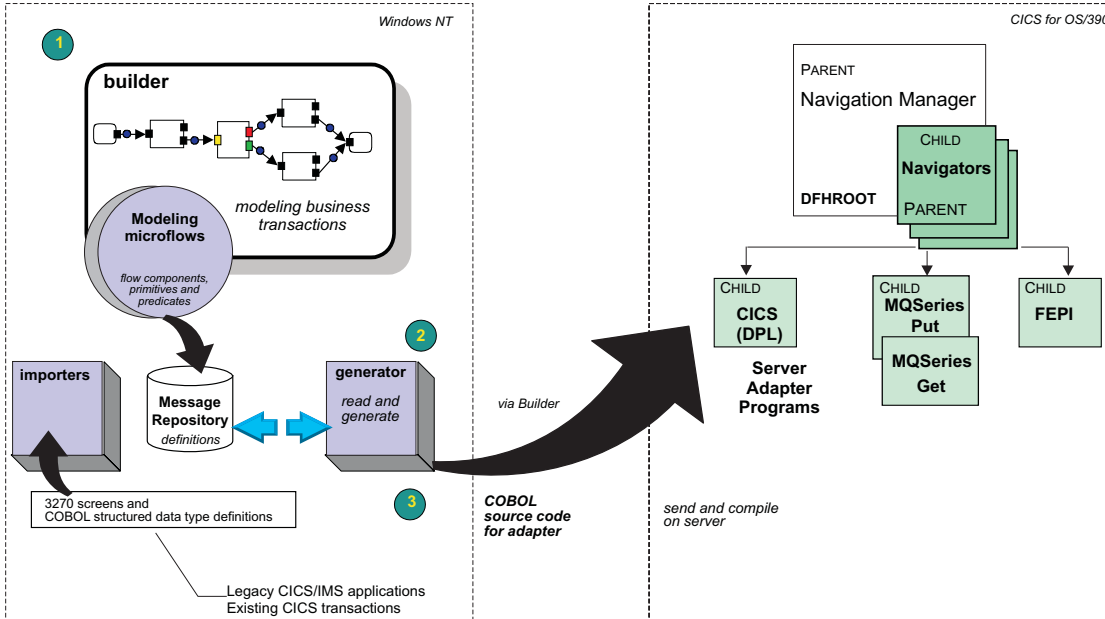
Building an adapter

1. Use builder to model a microflow representing the required adapter behavior.

2. Generate COBOL source code for microflow.

3. Transport the COBOL source code to an OS/390 server for compilation.

Executing an adapter

4. Controlling application initiates model execution (adapter request processing) at run time.

5. The DPL MQSI Agent for CICS Stub program uses information in the request message to:

   a. Read the Properties file

   b. Define the BTS process

   c. Write containers and run the BTS process

   d. Initiate the programmatic functions (adapter request processing) that enable the business transaction to be processed.

Run time processing

Depending on the microflow model, server adapter program processing can occur with:

- legacy CICS/IMS applications
- CICS transactions
- MQSeries-enabled applications
- custom programs

If modeled, the FEPI server adapter program might perform screen navigation.

The following list describes the processing that can be performed by the four types of server adapter programs:

1. Initiate programs via one or a sequence of DPLs, via CICS LINK.
2. Perform screen navigation. MQSI Agent for CICS conducts an interactive 3270 request and reply dialog with legacy CICS/IMS applications. Using IBM's FEPI product, it sends requests to and receives replies from any CICS/IMS application whose 3270 datastream is intended for a SLU2 3278 Model 2 terminal (24 rows by 80 columns), that is, the buffer in a single send/receive is not greater than 3600 bytes.

   MQSI Agent for CICS:

   - Begins the FEPI session
   - Parses screens sent by the legacy CICS/IMS application
   - Identifies the screen and its fields, attributes and data
   - Constructs and sends an appropriate reply, based on the modeling and on simple business logic,
   - Handles the next screen by parsing, identifying and constructing a reply or keystroke, and so on
   - Manages state information about the status of LUs, and
   - Ends the FEPI session.

   Screen navigation is modeled at build time.
3. Perform MQSeries PUT and GET commands in a synchronous fashion. In other words, in response to each PUT, MQSI Agent for CICS expects one reply.
4. Initiate custom programs via CICS LINK. Custom programs can contain complex rules such as logic and complex I/O. A custom program can be developed to augment the MQSeries Integrator Agent for CICS.

   Note: The mechanism to invoke the custom program is exactly the same as the mechanism to invoke a server adapter program with CICS transactions, that is, DPL.)

See "Adapter request processing" for a detailed scenario and illustration of how an adapter executes to process a request from a controlling application.

Adapter request processing

The following information explains the processing sequence (in synchronous mode) that occurs when the controlling application invokes the server run time by sending a request message. For an illustration of this process, see Figure 3 on page 7.

1. The controlling application invokes server run time by sending a request message.

- The controlling application is responsible for ensuring that the request message is translated, mapped, formatted and set to indicate whether a reply message is expected.
- The request message contains parameter values such as process name and process type that are required to initiate adapter request processing.

2. If the controlling application invokes the MQSeries Integrator Agent for CICS via the MQSeries product, the MQSeries-CICS Bridge Link task links to the DPL Stub program with a CICS communication area (COMMAREA). If the controlling application is not using the MQSeries product, it relies on a CICS-supplied interface to link to the DPL Stub program with a CICS communication area (COMMAREA).

3. The DPL Stub program performs the following functions:
   - Defines the process.
   - Creates the containers.
   - Runs the BTS process synchronously (an instance of the MQSeries Integrator Agent for CICS).

4. Based on the request (as denoted by the request name parameter in the DFHMAHV of the request message) sent by the controlling application, the Navigation Manager:
   - Defines the appropriate Navigator activity (BTS activity).
   - Runs the Navigator synchronously.

5. The Navigator:
   - Executes the scripted server adapter programs in accordance with the microflow model.
   - Waits for the adapter request processing to complete.

6. Depending on the microflow model, server adapter program processing can occur with the legacy CICS/IMS applications, CICS transactions, MQSeries-enabled applications and custom programs. If modeled, the FEPI server adapter program might perform screen navigation.

   Data is read from and written to the data-containers.

7. At any time, the Navigation Manager, Navigators and server adapter programs, if need be, write error records to an MQSeries queue.

8. The Navigation Manager updates its output data-container with the adapter reply message and returns with the adapter reply message to the DPL Stub program, which in turn returns to the controlling application.

9. If via the MQSeries product, the MQSeries-CICS Bridge Link task responds to the controlling application. If the MQSeries producct was not used, a CICS-supplied interface responds to the controlling application.
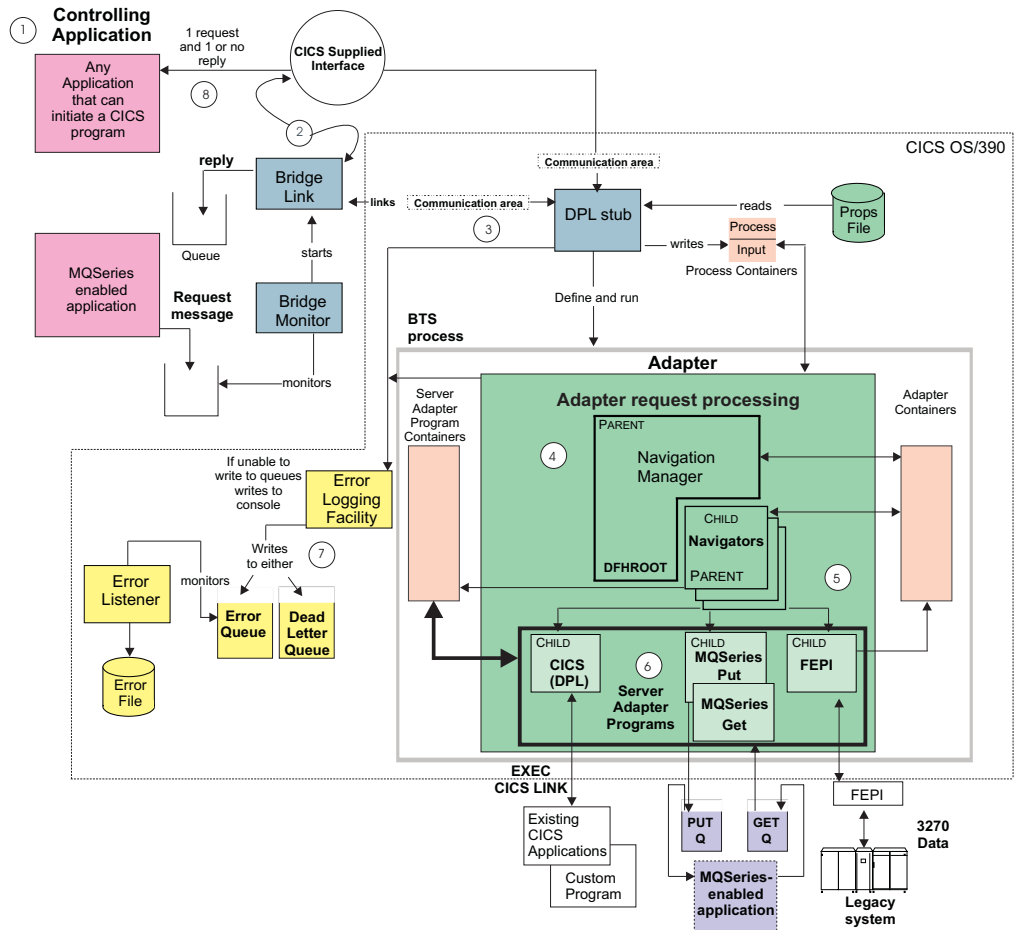
Figure 3. Run time interior — synchronous mode

Chapter 2. Benefits of MQSeries Integrator Agent for CICS Transaction Server

MQSeries Integrator Agent for CICS has the following benefits:

- MQSeries Integrator Agent for CICS is a member of IBM's common tooling for business integration that provides a define-once and deploy-many capability for process and message flows.

- MQSeries Integrator Agent for CICS enables more efficient use of computing resources. It off-loads work from the controlling application. At run time, instead of a controlling application invoking each transaction individually, it can invoke adapter request processing that can:
  - Invoke the CICS/IMS transactions, legacy CICS applications or MQSeries-enabled applications, and
  - Handle all of the request processing.

- MQSeries Integrator Agent for CICS uses CICS business transaction services (BTS) that simplifies the development of business transactions.

- MQSeries Integrator Agent for CICS stores models in XML in a common message and flow dictionary. XML is a standard for data representation.

- MQSeries Integrator Agent for CICS can be deployed without changing applications or business processes at all. Typically, all the integration work is performed in MQSeries Integrator Agent for CICS.

- A common interface for messaging enables adding new applications without changing existing applications or business processes.

- MQSeries Integrator Agent for CICS can reduce the need to write custom code.

Notices

This information was developed for products and services offered in the United States. IBM may not offer the products, services, or features discussed in this information in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this information. The furnishing of this information does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other 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 NON-INFRINGEMENT, 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.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Notices

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England
SO21 2JN.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Trademarks

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

CICS	IBM	MQSeries
FEPI		

Lotus and LotusScript are trademarks of Lotus Development Corporation in the United States, or other countries, or both.

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

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

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

IBM®

Printed in U.S.A.