

SupportPac CB12



General Insurance Application (GENAPP) for IBM CICS Transaction Server

SupportPac CB12



General Insurance Application (GENAPP) for IBM CICS Transaction Server

Contents

The general insurance application 1
Application architecture 1
Installing the general insurance application 4
Building the application environment 6
Testing the general insurance application 8

General insurance application reference 9
Notices 13
Trademarks 14

The general insurance application

The general insurance application is a working COBOL application that you can use to try out different features of CICS, including modernizing applications.

The general insurance application simulates transactions made by an insurance company to create and manage its customers and insurance policies. The application provides sample data and a 3270 interface for creating and inquiring on customers and policy information. Because the application is designed to simulate the flow of an application, some aspects of the application architecture do not use best practices. However, the application is designed to be extended to demonstrate other ways of accessing and modernizing traditional applications that are best practices.

You can use the general insurance application to try the following CICS features:

- Creating a topology that is managed by CICSplex SM
- Creating web services
- Creating workload simulations with Workload Simulator
- Creating business events and viewing them through a Web 2.0 dashboard (dynamic scripting application)

The SupportPac documentation describes the application architecture, how to install and set up the application, and how to test the application is working correctly. When you have a working application, you can try out these features of CICS. Each feature is described as a scenario in the CICS TS V4.2 Information Center.

Application architecture

The general insurance application is a 3270 application that runs in a single region. The application uses a named counter server and temporary storage queues in the coupling facility on z/OS. It writes to a VSAM file and DB2. As you start to use the application to try different features of CICS, the configuration of the application changes to include additional components.

The application is summarized in the following diagram:

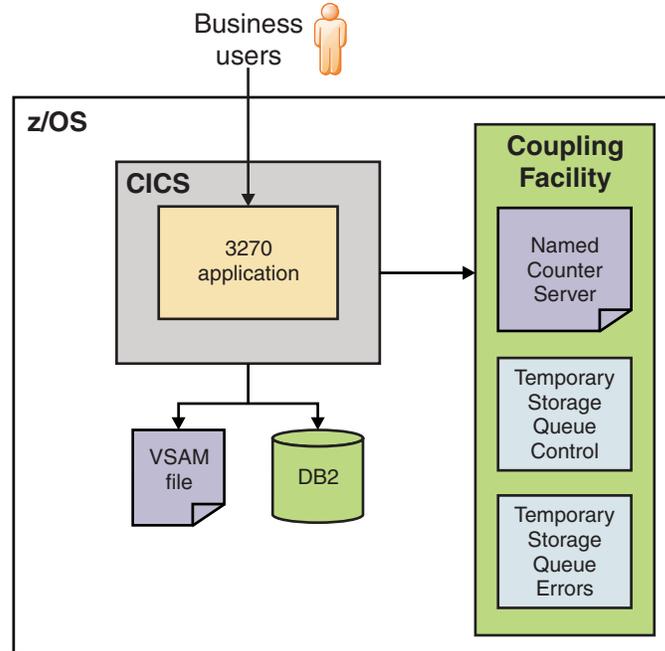


Figure 1. Initial CICS application configuration

Business users

The application has a 3270 interface that can be accessed from a terminal. The application uses a BMS map to control the screen layout for the 3270 interface. The application has five transactions that you can run to perform different tasks in the application:

- Run SSC1 to inquire on existing customer records and add customer records
- Run SSP1 to create a motor insurance policy
- Run SSP2 to create an endowment insurance policy
- Run SSP3 to create a house insurance policy
- Run SSP4 to create a commercial property insurance policy

The TRANSACTION resource definitions are supplied for you. Each transaction runs a program that provides the presentation logic for the application.

3270 application

The application programs are written in COBOL and are split into presentation logic, business logic, and data management logic. Designing the application in this way is a best practice and means that the application can be easily componentized to run in a topology under the control of CICSplex[®] SM.

When a business function is selected from the application menu in the presentation logic, the program uses an **EXEC CICS LINK PROGRAM** API command to call the correct program that contains the business logic. In the business logic, the program links to the program that contains the data management logic by using the same API command. The data management program takes control and performs updates to the DB2 database and VSAM file, before returning to the business logic. The business logic layer returns to the presentation logic layer and the user.

When you add a customer record, the program that drives the business logic gets a unique customer number from the named counter server by using an **EXEC CICS GET COUNTER** API command. When you add a policy, the policy number is generated by DB2 when the table is defined. The number is set by an SQL **INSERT** command.

The application also updates temporary storage queues in the coupling facility. These are not required by the application, but are provided for workload testing.

The application uses a two-phase commit to store customer and policy data in DB2 and, if successful, it updates VSAM. The requirement for a VSAM file is not a best practice, but is provided as a way to test two-phase commit processing in CICS.

Named counter server

To support adding customer records to the database, the application uses a named counter server in the coupling facility to provide unique customer numbers. A named counter can generate unique sequence numbers for applications. The advantage of using this approach is that all CICS regions in a Parallel Sysplex have access to the same named counters. However, if your application uses DB2, you can use the features of DB2 to create unique customer numbers for each record as they are added to the database.

Temporary storage queue (control)

The temporary storage queue, GENACNTL, contains a record of the range of customer numbers that are allocated. The application updates the temporary storage queue (control) as each customer record is added. Using a temporary storage queue to contain this type of data is not a best practice, but is provided for Workload Simulator automation.

Temporary storage queue (errors)

The temporary storage queue, GENAERRS, contains errors when the application is unable to write information to DB2. Using a temporary storage queue in this way is not a best practice, but is used to easily identify any DB2 problems during your testing.

Coupling facility

The coupling facility is part of z/OS and contains the named counter server and temporary storage queues that are used by the application. The coupling facility contains a set of structures for the named counter server and temporary storage queues. It contains a pool that associates the structures with the CICS region.

The coupling facility provides a mechanism to share data across logical partitions on a z/OS system. All CICS regions operating can access the data, even if they are in different partitions. Although using the coupling facility for an application in a single region is not typical, the application is designed to run in a topology and therefore the coupling facility is used.

VSAM file

The application stores information in VSAM files. The general insurance application uses two VSAM files: KSDSCUST for customer records and KSDSPOLY for policy records. You can browse these files in the *userid.CB12.KSDSCUST* and *userid.CB12.KSDSPOLY* data sets to look at the contents. The key for the KSDSCUST VSAM file is the first 10 characters of each customer record. The key for the policy VSAM file is the first 21 characters. The application uses four types of policies, indicated by the initial letter:

- C is a commercial property insurance policy
- E is an endowment insurance policy
- H is a house insurance policy
- M is a motor insurance policy

The next 10 characters are the customer ID, followed by 10 characters for the policy number. For example, M00000000050000000003DENNIS is a motor policy, where the customer number is 5, the policy number is 3, and the make of car is Dennis.

DB2 The database contains details of the customers and insurance policies for the application. The database contains the following tables:

- A customer table that lists all the customer records, including the customer number.
- A policy table that lists all the policies, including the customer number, policy number, and policy type.
- A policy table for each type of policy: commercial, endowment, house, and motor insurance policies.

These tables are related for referential integrity; for example, you cannot create a policy if the customer record does not exist. CICS uses the DB2 attachment facility to access the tables in the database to add and inquire on customer and policy information.

CICS region

The CICS region is configured with a set of system initialization parameters that enable CICS to connect with DB2 and the coupling facility. The region also contains the resources required by the application, including a TSMODEL resource with a pool name that matches the structure in the coupling facility, a DB2CONN resource for the connection to DB2, and FILE resources. For a complete list of the CICS resources that are supplied with the application, see “General insurance application reference” on page 9.

Installing the general insurance application

The general insurance application is a free download from the IBM website and contains the jobs you need to set up the application.

Before you begin

The general insurance application requires a CICS region as a minimum, but can be extended to run in a CICS topology and CICSplex. You must also have a supported version of DB2 and a COBOL compiler installed.

About this task

The application is packaged in a .zip file. You must extract the file, and use FTP to copy it to your z/OS system. The files include a customization job so you can easily configure the application for your environment. The following files are included in the .zip file:

Table 1. List of extracted files

File	Description
CNTL.XMIT	JCL to build the environment for the application

Table 1. List of extracted files (continued)

File	Description
EXEC.XMIT	REXX code that customizes the JCL
KSDSCUST	Sample customer information for the application
KSDSPOLY	Sample policy information for the application
SOURCE.XMIT	Source programs for the general insurance application
Transaction_Counters.evbind	Event binding for the general insurance application
WSIM.XMIT	Sample workload simulator scripts

Procedure

- Use FTP to transfer the files to the z/OS system and put them in MVS data sets. Call the data sets `CB12.filename`, where *filename* is the name of the extracted file; for example, `CB12.SOURCE.XMIT`.
 - Transfer the files with the suffix XMIT in binary mode. Enter `bin` on the command line to change to binary mode.
 - Transfer the KSDSCUST file in ASCII mode. Enter `ascii` on the command line to change to ASCII mode.
 - Set the record length of the data set for the KSDSPOLY file before transferring it. Enter the command `quote site lrecl=64 blksize=6400` to set the record length.
 - Transfer the KSDSPOLY file in ASCII mode.
 - Do not transfer the event binding to the z/OS server, because this file is imported into CICS Explorer.
- Extract the SOURCE, WSIM, CNTL, and EXEC files into partitioned data sets by using the **RECEIVE** command:

```
RECEIVE INDSN('userid.CB12.SOURCE.XMIT')
```

This command identifies the input data set. You must enter a response to the **RECEIVE** command to identify the name of the destination partitioned data set, as shown in the following example:

```
DA('userid.CB12.SOURCE')
```

You do not have to perform this step for the KSDS files because they are already in a readable format.

- Customize the member CUST1 in the `CB12.EXEC` data set. The values in this member include settings for a CICS topology and CICSplex SM, which you can ignore. Replace the following values with your own:

```
PDSMEMin = 'userid.CB12.CNTL'
CICSHLQ   = 'CTS420.CICS670'
CPSMHLQ   = 'CTS420.CPSM420'
CSDNAME   = 'userid.GENAPP.DFHCS'D
USRHLQ    = 'userid'
COBOLHLQ  = 'PP.COBOL390.V420'
DB2HLQ    = 'SYS2.DB2.V910'
CEEHLQ    = 'CEE'
DB2RUN    = 'DSN910PM'
SQLID     = 'STTESTER'
DB2SSID   = 'DHM1'
DB2DBID   = 'GENASA1'
WSIMHLQ   = 'WSIM.V110'
```

PDSMEMin

Enter the location of the CNTL library, *userid.CB12.CNTL*. Replace *userid* with your user ID.

CICSHLQ

Enter the CICS high-level qualifier to customize the CICS data sets.

CPSMHLQ

Enter the CICSplex SM high-level qualifier to customize the CICS data sets.

CSDNAME

Enter the fully qualified name of the CSD for the CICS regions that run the general insurance application.

USRHLQ

Enter a high-level qualifier for the application data sets.

COBOLHLQ

Enter the high-level qualifier for the COBOL compiler.

DB2HLQ

Enter the high-level qualifier for DB2 libraries.

CEEHLQ

Enter the high-level qualifier for Language Environment.

DB2RUN

Enter the high-level qualifier for the DB2 runtime library.

SQLID

Enter the RACF user ID that is authorized to create objects in DB2.

DB2SSID

Enter the subsystem ID of the DB2 instance that you want to use.

DB2DBID

Enter a name for the database that is going to contain the general insurance application data. You can use any value.

WSIMHLQ

Enter the high-level qualifier for the Workload Simulator for z/OS libraries. If you do not use this product, you can ignore this parameter.

4. Run the EXEC job to customize the JCL in the CNTL data set. The job copies the members and updates them with your values. Each member name is prefixed with @ to indicate that it is customized. You can run the customization step as many times as you want.

Results

You successfully installed and customized the general insurance application jobs.

What to do next

The next step is to build the environment for the application.

Building the application environment

Run the customized jobs to create the DB2 database, application files, and CICS resources.

About this task

To set up the general insurance application, run each of the customized jobs in the CB12.CNTL data set.

Procedure

1. Submit the job @ADEF121 to create the VSAM application files for customer details and policy details. This job also populates the KSDSCUST and KSDSPOLY files with data.
2. Submit the job @ASMMAP to build the BMS maps for the 3270 interface. This job has a return code of 4.
3. Submit the job @CDEF121 to add all of the required resource definitions to the CSD. The group list is called GENALIST.
4. Submit the job @COB0L to compile the COBOL application programs. The compiled programs are put in the *USERHLQ.CB12.LOAD* library.
5. Submit the job @DB2CRE to create the DB2 database for the application. It creates a storage group, database, tables, and indexes. This job also populates the database with data.
6. Submit the job @DB2BIND to bind the application to the DB2 objects.
7. Create a structure in the coupling facility for a named counter server. The pool name is GENA and the structure name is DFHNCLS_GENA. For details on how to define a list structure for a named counter server, see *Setting up and running a named counter server* in the *System Definition Guide*.
8. Create a structure in the coupling facility for a shared temporary storage queue. The shared temporary storage queue is also called GENA and the structure name is DFHXQLS_GENA. For details on how to define a list structure for a temporary storage server, see *Setting up and running a temporary storage server* in the *System Definition Guide*.
9. Submit the job @SAMPNCS to create a named counter server called GENA. You can customize this sample to change the pooled name if appropriate. The job is long-running and might take a while to complete. You can check that the job completed successfully by looking in the system log for the CICS message DFHNC0102I Named counter server for pool GENA is now active.
10. Submit the job @SAMPTSQ to create a temporary storage queue server called GENA. Again you can customize the sample to change the pooled name if appropriate. This job is also long-running and might take a while to complete. You can check that the job completed successfully by looking in the system log for the CICS message DFHXQ0102I Shared TS queue server for pool GENA is now active.
11. Edit the startup JCL for the CICS region:
 - a. Update the system initialization parameters to provide the following nondefault values:

```
GRPLIST=(DFHLIST,GENALIST)
DB2CONN=YES
NCPLDFT=GENA
```
 - b. Add the load library name *USERHLQ.CB12.LOAD* library to the RPL.
12. Submit the job to start the CICS region.

Results

The CICS region starts up successfully. If you want to rebuild the environment, submit the job @DB2DEL to delete the DB2 database.

What to do next

Validate that the general insurance application works and can connect to DB2.

Testing the general insurance application

To validate that your setup is correct, run the general insurance application using its 3270 interface.

About this task

The application has several transactions that you can run to access customer information and policies.

Procedure

1. Run the LGSE transaction to build the control tables for the application. The transaction also clears out the shared temporary storage queue and the named counter server.
2. Run the SSC1 transaction to open the application customer menu.
3. In the **Cust Number** field, enter a number between 1 and 10 to browse the customer records and validate that the application can access the DB2 database. The application contains 10 customer records.
4. In the **Select Option** field, enter 1 to inquire on the customer. The customer information is returned from DB2.
5. Exit the application by pressing F3.
6. Run the SSC1 transaction again to add a customer to the database:
 - a. Enter the details for a new customer.
 - b. In the **Select Option** field, enter 2 to add the customer record to the database.

A unique customer number is allocated from the named counter server. The application adds the customer record to DB2 and the VSAM file in a two-phase commit. If the VSAM file update fails, the DB2 update is rolled back.

7. Run the SSP1 transaction to access the motor policy menu. To check that you can access policy information:
 - a. In the **Policy Number** field, enter 00000000001. In the **Cust Number** field, enter 0000000002.
 - b. In the **Select Option** field, enter 1 to return the motor policy information from the database.

Results

The application is successfully set up and running in a single CICS region. The application can access DB2 and write to VSAM files.

What to do next

You can try the scenario to run the insurance application in a CICS topology and manage it using CICSplex System Manager. For details, see Scenario: Creating a CICS topology that is managed by CICSplex System Manager.

General insurance application reference

The general insurance application supplies jobs, scripts, and programs to customize and extend the scope of the application to demonstrate different functions in CICS®. The supplied files are listed here as a reference.

Installation files

When you download the SupportPac and extract it, the follow files are supplied:

Table 2. List of extracted files

File	Description
CNTL.XMIT	JCL to build the environment for the application
EXEC.XMIT	REXX code that customizes the JCL
KSDSCUST	Sample customer information for the application
KSDSPOLY	Sample policy information for the application
SOURCE.XMIT	Source programs for the general insurance application
Transaction_Counters.evbind	Event binding for the general insurance application
WSIM.XMIT	Sample workload simulator scripts

Supplied jobs

The SupportPac supplies a set of jobs that you must customize to work in your CICS environment. The jobs are listed in the following table. Any job prefixed with @ is customized when you install the general insurance application.

Table 3. Customized jobs for the general insurance application

Member	Description
@ADEF121	Delete, define, and load application data sets
@ASMMAP	Assemble BMS maps
@CDEF121	CICS resource definitions for single region
@CDEF122	CICS resource definitions for topology environment
@CDEF123	CICS resource definitions for workload managed environment
@CDEF124	CICS resource definition for web services
@CDEF125	CICS resource definition for workload simulation
@COBOL	Compilation job for application programs
@CPSMDE2	CICSplex SM definitions
@DB2BIND	Bind job for DB2 application
@DB2CRE	Create DB2 objects to support application
@DB2DEL	Delete existing GENAPP database
@DEFDREP	Define CMAS data repository data set
@DEFWREP	Define WUI repository data set
@ITPENTR	Start Workload Simulator to create customer and policy records
@ITPLL	Format Workload Simulator log file
@ITPSTL	Translate Workload Simulator programs into scripts
@SAMPCMA	Sample JCL to start a CMAS

Table 3. Customized jobs for the general insurance application (continued)

Member	Description
@SAMPNCS	Sample JCL to start a named counter server
@SAMPTSQ	Sample JCL to start a shared temporary storage queue server
@SAMWUI	Sample JCL to start the CICSPlex SM Web User Interface
@WSAAC01	Sample JCL for web services assistant to create web service for customer add function
@WSAAP01	Sample JCL for web services assistant to create web service for policy add function
@WSAIC01	Sample JCL for web services assistant to create web service for customer inquiry function
@WSAIP01	Sample JCL for web services assistant to create web service for policy inquiry function
@WSAVC01	Sample JCL for web services assistant to create web service for reading VSAM file function

Application transactions

The general insurance application has transactions for starting application menus and internal transactions for processing the business logic. The following transactions are supplied:

Table 4. Transactions that are supplied with the general insurance application

Name	Description
LGSE	Initialize counters and temporary storage queues
LGCF	Retrieve random customer number from VSAM file
LGPF	Retrieve policy and customer number from VSAM file
LGST	Event adapter trigger to update counters
SSC1	Menu for customer details
SSP1	Menu for motor policy options
SSP2	Menu for endowment policy options
SSP3	Menu for house policy options
SSP4	Menu for commercial property options
SSST	Initialize and copy values for dynamic scripting application

Application programs

The general insurance application comprises the following programs:

Table 5. Programs that are supplied with the general insurance application

Name	Description
LGACDB01	Add customer details to DB2® database
LGACUS01	Add customer business logic
LGACVS01	Add customer details to VSAM file
LGAPDB01	Add policy details to DB2 database
LGAPOL01	Add policy business logic

Table 5. Programs that are supplied with the general insurance application (continued)

Name	Description
LGAPVS01	Add policy details to VSAM file
LGASTAT1	Update transaction counts by using named counters
LGDPDB01	Delete policy record from DB2 database
LGDPOL01	Delete policy business logic
LGDPVS01	Delete policy record from VSAM file
LGICDB01	Retrieve customer details from DB2 database
LGICUS01	Inquire customer business logic
LGICVS01	Retrieve customer record from VSAM file
LGIPDB01	Retrieve policy record from DB2 database
LGIPOL01	Inquire policy business logic
LGIPVS01	Retrieve policy record from VSAM file
LGSETUP	Initialize counters and temporary storage queues
LGSTSQ	Write a message to a temporary storage queue
LGTESTC1	Presentation logic for customer menu
LGTESTP1	Motor policy presentation logic
LGTESTP2	Endowment policy presentation logic
LGTESTP3	House policy presentation logic
LGTESTP4	Commercial property policy presentation logic
LGUPDB01	Update DB2 policy record
LGUPOL01	Update policy business logic
LGWEBST5	Copy business transaction counts from named counters to temporary storage queue
SSMAP	BMS map screen layout for presentation menus

Other resources

The SupportPac supplies definitions for other resources that are required by the general insurance application:

Table 6. Other resources that are supplied with the general insurance application

Name	Resource type	Description
DB2SSID	DB2CONN	Attachment definition for the DB2 connection
GENAMOD	ENQMODEL	Enqueue that is used for updating shared temporary storage queues
GENAEV01	BUNDLE	Bundle that contains event binding and adapter definitions
GENAPIP1	PIPELINE	Provider mode pipeline for processing web service requests
GENASTST	TSMODEL	Model for access to shared temporary storage queues
GENATCP1	TCPIPSERVICE	TCP/IP connection that is used for web services
KSDSCUST	FILE	VSAM file definition for customer records
KSDSPOLY	FILE	VSAM file definition for policy records

Notices

The provisions set out in the following two paragraphs do not apply in 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.

Information contained and techniques described in this publication have not been submitted to any formal IBM test and are distributed on an "AS IS" basis.

The use or implementation of any information contained and/or of any technique described in this document is the user's responsibility and depends on the user's ability to evaluate and integrate the information and/or technique into the user's operational environment. While IBM has reviewed each item for accuracy in a specific situation, IBM offers no guarantee or warranty that the same or similar results will be obtained elsewhere. Users attempting to adapt any technique described in this document to their own environments do so at their own risk.

The information contained in this publication could include technical inaccuracies or typographical errors.

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

Any reference in this publication to an IBM licensed program or another IBM product 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 applicable intellectual property rights may be used instead of the referenced IBM licensed program or other IBM product.

The user is responsible for evaluating and verifying the operation of the material supplied in conjunction with this publication in conjunction with other products, except those expressly designated by IBM.

International Business Machines Corporation may have patents or pending patent applications covering subject-matter described in this document. The furnishing of this document does not give you any license to any such patent. You can send license inquiries, in writing, to:

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.

