

# IBM CICS Client for AIX - Sample Programs README

## Contents

The CICS Server Programs  
The IBM CICS Universal Client Programs  
Program Descriptions

## The CICS Server Programs

To successfully run the sample programs you will need to have the correct server programs and transactions built and available on your CICS server. The required sample server programs can be found in the client samples/server directory under your root installation directory. Four COBOL programs are provided:

- EC01.CCP is the first sample ECI backend program. It simply returns the current date and time in its COMMAREA.
- EC02.CCP is the second sample ECI backend program. It simply returns the number of times it has been run in a Unit of Work in its COMMAREA.
- EP01.CCP is the first sample EPI transaction for use with the C, COBOL and PL/1 samples. It simply returns the number of times it has been run as the contents of a 3270 datastream.
- EP02.CCP is the second sample EPI transaction for use with the C++ and COM samples. It simply returns the number of times it has been run as information in fields of a BMS map.
- EP02MAP.BMS is the map source for use with EP02.

For information about how to build and install these programs you should refer to your CICS server documentation.

## The IBM CICS Universal Client Programs

The IBM CICS Universal Client provides sample ECI and EPI programs written in C, C++ and COBOL. These samples cover the two API's available to the CICS Universal Client for AIX programmer:

Standard (c) API

C++ Class Libraries API

### *Standard API Samples*

To build the samples you must use the supplied makefiles found in each of the language directories. i.e. make -f csamp.mak

### *C++ Class Libraries API Samples*

To build the C++ samples you must use the makefile provided in the samples\cpp directory. i.e. make -f cppsamp.mak

IBM VisualAge C++ v4.0 users please note that to build the C and C++ samples they must use the configuration files (.icc) provided.

## Program Descriptions

### *Basic ECI - ECIB1*

This sample simply lists the systems defined in ctg.ini and allows the user to choose the one to which an ECI call is made. This call is then made and the date and time are returned by transaction EC01.

### *Intermediate ECI - ECII1*

This sample lists the systems defined in ctg.ini and allows the user to choose the one to which an ECI call is made. A unit of work is then started and the first ECI call is made to EC02. In this sample all calls are Asynchronous and are handled using event functions that are found in the supplied "cclcalls.h" header file. This file must be in your include path when building the samples. The user is then asked if they wish to run EC02 again or finish the Unit of Work. On exit from the loop the user can choose to either Commit or Backout the current Unit of Work.

### *Advanced ECI - ECIA1*

This sample uses multithreading to call EC01 on every server defined in your "ctg.ini" file. Parameters are passed to define the number of threads run for each server, the number of calls to EC01 made per thread and the type of calls made i.e. Synchronous or Asynchronous. Again this thread needs the supplied header file "cclcalls.h" for the threading functions used.

### *Basic EPI - EPIB1*

This sample simply lists the systems defined in ctg.ini and allows the user to choose the one to which an EPI call is made. This call is then made and the 3270 datastream returned by the transaction EP01 is output to screen.

### *Intermediate EPI - EPII1*

This sample lists the systems defined in ctg.ini and allows the user to choose the one to which an ECI call is made. A terminal is then installed Asynchronously and transaction EP01/2 is called asynchronously. Once a reply is recieved the screen returned by EP01/2 is returned and the user is prompted to continue or quit. The header "cclcalls.h" will again be needed for the event handling functions within this sample.

### *Advanced EPI - EPIA1*

This sample uses multithreading to call EP01/2 on every server defined in your "ctg.ini" file. Parameters are passed to define the number of threads run for each server, the number of calls to EP01/2 made per thread and the type of calls made i.e. Synchronous or Asynchronous. Again this thread needs the supplied header file "cclcalls.h" for the threading functions used.

### *Basic ESI - ESIB1*

This sample simply lists the systems defined in ctg.ini and allows the user to choose the one to which an ECI call is made and which their security is verified on. A userid and pasword are then entered for verification on the chosen server using the new ESI API.

Once verified the userid and password are used in a synchronous ECI call of EC01. The reply of EC01 is then printed to screen.