ZCICSSYS.TXT

Z390/CICS System Programmer's Guide

This document describes the Z390/CICS environment, how it operates and the modifications that can be made to it.

Experienced CICS people may notice a lack of authenticity in the background coding and formats, and some use of archaic methods.
This is all intentional, and as Z390 matures, so will Z390/CICS.

Objectives

1) To take an existing Assembler CICS application, re-assemble it in the Z390 environment and run it unmodified and successfully under Z390/CICS.

2) To be able to develop an Assembler CICS application in the Z390 environment and transfer the source to the mainframe for re-assembly and testing.

Only the source may be exchanged between any environments.

Version compatability

Versions of Java and Z390 are tested by Don and there are mechanisms for checking compatability.

In Z390/CICS History each Z390/CICS release has a Z390 version.
This is the release/PTF that was used to test Z390/CICS.
Previous releases of Z390 may not work.

Future releases of Z390 will work unless there is a PTF that requires source modification or re-assembly. When that happens, I will ask Don to add an instruction to the README and provide fixes as needed.

Current environment

It is fair to say that the facilities available to Z390/CICS Application programmers are somewhat limited at present. These are early days, and functions will be added gradually.

The currently supported Application environment is described in the Z390/CICS Application Programming Guide.

Requests for commands and extra parameters are very welcome and will help to set a priority list.

Re-entrancy

Much of the internal code does not conform to the strict (quasi)-re-entrancy rules required of mainframe CICS.

If you are developing an application for later transfer to the mainframe, you must obey all the re-entrancy rules. There is no checking yet for rule breaking.

How it works

I have split the Z390/CICS environment into two sections.

The primary task is Z390CICS, which I have called the Global Manager. This will handle all shared resources like TS, FC, etc.

Each terminal has its own Command Prompt (MS-DOS task).
This environment runs Z390KCP and invokes any Application programs requested.

Each Command Prompt is effectively a single terminal, single task environment.

Z390KCP is therefore the Local Manager handling EIB, COMMAREAs, DSAs and other task related storage.

TCPIO SEND/RECEIVE are used to pass requests and data between Z390CICS and each Z390KCP, with Z390CICS being the server and all terminals running Z390KCP as multiple clients.

Setting it up

Parameters...

   The Z390CICS.INI file is self-documenting.

The following BAT files may need to be modified to your own environment...

   Z390CICG - Start up CICS

   Z390KCPR - Start 1 remote terminal for testing/tracing.

   Z390KCPL - Start many remote terminals.

DFHPCT.MLC has a basic set of test transaction codes which are listed later. Users should add their own transactions to DFHPCTUS.CPY and re-assemble the PCT.

DFHFCT.MLC has a basic set of test files. Users should add their own files to DFHFCTUS.CPY and re-assemble the FCT.
File creation and setup is more fully described in the Z390/CICS VSAM Guide.

Local and Remote terminals

A Local terminal is a Command Prompt (MS-DOS task) that is auto-started when Z390CICS is started. The number of Local terminals that are started is controlled by the LOCAL_TERMINALS= parameter.

A Remote terminal must be started from a Command Prompt that you have manually created and is initiated by using Z390KCPR or Z390KCPL (see the 'Starting it up' section).

A Remote terminal doesn't have to be on the same PC as Z390CICS, but can be on another PC connected via a home network.

There is a restraint on the number of Remote terminals set by the REMOTE_TERMINALS= parameter.

The MAXTHREADS= parameter defines the upper limit for all terminals.

Starting it up

From the Z390 GUI (recommended), or in your own Command Prompt...

1) Z390CICG
This will start the Z390/CICS Global Manager and all Local terminals.

2) A Remote terminal may be set up as follows...

a) Method 1 : One remote for testing
If interactive debugging of an application is wanted...
Create a Command Prompt window and use CD to navigate to, and invoke Z390KCPR. Parameters like TEST and TRACE may be added.
Any test session will be recorded in Z390KCP.TRE.

The remote terminal will be started in the Command Prompt.

If TEST is specified, then Z390/CICS progress cannot be made unless you give commands to the Z390 GUI as well.

b) Method 2 : The Launcher
Create a Command Prompt window and use CD to navigate to, and invoke Z390KCPL.

Z390KCPL takes a single digit parameter, eg. Z390KCPL 2
A parameter of 1 is assumed if missing.

Z390KCPL does not use the current Command Prompt, but creates the specified number of remote terminals in other windows.

The use of TEST or TRACE is not recommended as they will only affect the launcher program Z390RMTE.

The number of terminals actually started is subject to both the REMOTE_TERMINALS= and the MAXTHREADS= parameters.

Ok, I've got a blank Z390/CICS screen or a logo...now what ?

When the terminal is opened, the termid is in the title.

From the 'initial screen' you can perform the following tests...

CTRL+C	Clear the screen. Within a transaction that has issued a RECEIVE, the CLEAR AID is returned.
AAAA	test invalid transid message
MMM1	test abend APCT message
MMM2	'hello world'
GUI4	conversational test 1
GUI6	conversational test 2
TST1	conversational test 3 and LINK test
BED1	Test bed for LINK, XCTL and RETURN with COMMAREA. Keep pressing ENTER until the 'clear screen' message is displayed.
BED2	HANDLE AID testing Follow the on-screen instructions. DUMP TRANSACTION testing.

```
   BED3        HANDLE CONDITION/IGNORE CONDITION/PUSH/POP testing
               GETMAIN/FREEMAIN testing
               Abend handling
               Follow the on-screen instructions.

   BED4        HANDLE ABEND testing. Simple handling.
               Follow the on-screen instructions.

   BED5        HANDLE ABEND testing. Complex handling.
               Follow the on-screen instructions.

   BED9        Temporary Storage testing.
               Also builds an environment for testing CEBR.
               Can be run multiple times to extend the CEBR test
queues.

   IC01        Test Interval Control functions.
               This task may take up to 2 mins to complete.

   VSM1        Read and browse of ESDS files.
               Records successfully read are written to the TS queue
               VSM1.

   VSM2        Read and browse of RRDS files.
               Records successfully read are written to the TS queue
               VSM2.

   VSM3        Read and browse of KSDS files.
               Records successfully read are written to the TS queue
               VSM3.

   BMS1        Test all functions of mapping support.

Shutdown
--------
CEMT P SHU IMM may still leave a Z390/CICS window behind.
I am working to resolve this, just click close (big X) on each
stranded window.

   CEMT S TER OUT
      This closes the terminal.
      Local terminals may not be re-instated.
      Even if all terminals are closed with this method, Z390/CICS
      will not shut down. A remote terminal may still be started.
```

```
    CEMT P SHU
       The terminal issuing this command will be closed.
       No new terminals or tasks may be started.
       If there are no active tasks, then the server is closed and
       Z390/CICS ends.
       When all active tasks terminate, Z390/CICS will end.

    CEMT P SHU IMM
       The terminal issuing this command will be closed.
       The server is then closed and Z390/CICS ends.
       Terminals with active tasks may remain stranded.
```

Abends

SNAP dumps are provided when requested or when circumstances demand them.

A standard abend message (DFH2206) is usually displayed.

An ASRA abend which is handled by a HANDLE ABEND command will always produce a SNAP dump but no message.

If the initial program of a transaction is not available an APCT abend will occur. This won't produce a dump and cannot be HANDLEd.

The ID of the SNAP indicates its origin...

```
   999 : Abend ASRA   TEXT='ABEND ASRA'
   998 : Requested dump by EXEC CICS ABEND  TEXT='ABEND abcode'
   997 : Requested dump by EXEC CICS DUMP
         COMPLETE     TEXT='DUMP dumpcode COMPLETE'
         SIngle area  TEXT='DUMP dumpcode AREA'
         Segments     TEXT='DUMP dumpcode SEGMENT nnn'

   Other values : The EIBRESP field, ie. 27=PGMIDERR has occured
                  TEXT='ABEND xxxx' ie. AEI0=PGMIDERR
```

Aborting the environment violently

This is occasionally necessary to preserve traces and dumps.

The Z390 GUI can stay up and won't be harmed by this process.

a) Right click the taskbar
b) Select Task Manager
c) Search for java.exe
   There might be more than one, repeat d) and e) below for each one
   You can sort the tasks by clicking on 'Image name'
   Don't confuse this with javaw.exe
d) Right click on java.exe
e) Select End Process and Yes.
   All the Z390/CICS environment and terminals should go away.

Change Summary

June 27, 2008
   Added BMS1 to list of test transactions

Trademarks

IBM, CICS and VSAM are registered trademarks of International Business Machines Corporation.

Author: Melvyn Maltz
Shipping Date: June 27, 2008
Z390 version: V1.4.02
Z390/CICS version: V5
→