

Z390/CICS Diagnosis Reference

=====

Index

Control Blocks... Client

- Dynamic Storage Area (DSA)
- EXEC Interface Block (EIB)
- HANDLE ABEND Block
- HANDLE AID Block
- HANDLE CONDITION Block
- Link-Level Area (LKA)
- Terminal Control Table-Terminal Entry (TCTTE)
- Temporary Storage request/reply block (DFHTSBLK)
- File Control request/reply block (DFHFCBLK)

Control Blocks... Server

- File Control Table (DFHFCT)
- VSAM Work Area (DFHVSAD)

Management descriptions...

- Abend
- COMMAREA
- GETMAIN/FREEMAIN
- HANDLE AID
- HANDLE/IGNORE CONDITION
- Temporary Storage
- File Control

Event Tracing

Internal Abends (Z390CICS)

Control Blocks... Client

Dynamic Storage Area (DSA).

- Eye catcher: None
- Acquired : DFHEI ENT
- Released : DFHEI RET
- Anchor : R13
- DSECT : DFHEI STG (prefix only)
- Cleared : Only the prefix, not the user area
- Length : Variable

EXEC Interface Block (EIB).

- Eye catcher: 'DFHEIBLK'
- Acquired : Z390KCP
- Released : Close of thread
- Anchor : R11 (DFHEIBR)
- DSECT : DFHEIBLK
- Cleared : Yes
- Length : EIBLENG

HANDLE ABEND Block.

- Eye catcher: 'DFHABBLK'
- Acquired : 1st use
- Released : Task end
- Anchor : TCTTEABD (Byte after eyecatcher)
- DSECT : DFHABBLK
- Cleared : Yes
- Length : ABDLENG (one table entry)

Entries : 25

HANDLE AID block.

Eye catcher: 'DFHADBLK'
 Acquired : HANDLE AID (1st block only)
 PUSH HANDLE
 Released : DFHEI RET (all chained AID blocks)
 POP HANDLE (only top-of-chain block)
 XCTL (all chained AID blocks)
 Anchor : DFHEI AID (DSA)
 Chain : AIDCHAIN
 DSECT : DFHADBLK
 Cleared : Yes
 Length : AIDLENG

HANDLE CONDITION block.

Eye catcher: 'DFHHCBLK'
 Acquired : HANDLE CONDITION (1st block only)
 PUSH HANDLE
 Released : DFHEI RET (all chained HANDLE CONDITION blocks)
 POP HANDLE (only top-of-chain block)
 XCTL (all chained HANDLE CONDITION blocks)
 Anchor : DFHEI HCN (DSA)
 Chain : HCNCHAIN
 DSECT : DFHHCBLK
 Cleared : Yes
 Length : HCNLENG

Link-Level Area (LKA).

Eye catcher: None
 Acquired : Z390KCP
 Released : Close of thread
 Anchor : TCTTELKA
 Cleared : Yes
 Length : 4 (R13 value for this Link-Level)
 Entries : 25

Terminal Control Table-Terminal Entry (TCTTE).

Eye catcher: 'DFHTCTTE'
 Acquired : Z390KCP
 Released : Close of thread
 Anchor : R10 (TCTTEAR)
 DSECT : DFHTCTTE
 Cleared : Yes
 Length : TCTTELEN

Temporary Storage request/reply block (DFHTSBLK).

Data sent/received follows the block.

Eye catcher: None
 Acquired : GETMAINd by EXEC CICS TS command
 Released : FREEMAINd by EXEC CICS TS command
 Anchor : None
 DSECT : DFHTSBLK
 Cleared : No
 Length : TSPREFIX

File control request/reply block (DFHFCBLK).

Data sent/received follows the block.

Eye catcher: None
 Acquired : GETMAINd by an EXEC CICS file control command
 Released : FREEMAINd by an EXEC CICS file control command
 Anchor : None
 DSECT : DFHFCBLK

Cleared : Yes
 Length : FCPREFIX

Control Blocks... Server

File Control Table (DFHFCT)

Contains the ACB for each file operation.

Eye catcher: None
 Acquired : Z390CICS at start
 Released : Never released
 Anchor : DFHFCTAD
 DSECT : DFHFCTDS
 Length : FCTABLEN

VSAM Work Area (DFHVSAD)

Also contains the RPL for file operations.

Eye catcher: None
 Acquired : Z390CICS at File Control process start
 Released : Z390CICS at File Control process end
 Anchor : FCTVSWA
 Chain : VSWCHAIN
 DSECT : DFHVSAD
 Cleared : Yes
 Length : VSWLEN

Abend Management

The following types of abend may occur...

Program check
 IGNORE CONDITION not permitted (program check)
 Condition raised but not handled or ignored
 EXEC CICS ABEND

Whether any of these result in a dump and/or termination of the task depends on the HANDLE ABEND status which is discussed later.

All of the above types cause a program check, with the following markers...

X' 000000', 'C' ABEND', A(IGNORE address)	-- HANDLE CONDITION
X' 0000FE', 'C' ABEND', 'C' xxxx'	-- ABEND with dump
X' 0000FF', 'C' ABEND', 'C' ????'	-- ABEND without dump
Other	-- Program check

The program check is trapped by the ESTAE routine APPABEND in Z390KCP. This routine determines the cause of the abend and takes the correct action...

HANDLE CONDITION marker...

The last CONDITION block is located (if any) and the condition slot is tested, followed by the ERROR slot.
 If no HANDLE or IGNORE then the terminate handler is invoked (see later).

ABEND with/without dump and program checks go straight to the terminate handler.

Terminate handler...

APPTABDN in Z390KCP tests for the existence of a HANDLE ABEND block (DFHABBLK, see above). If none the task is terminated abnormally.

ZCICSDRF.TXT

Each entry in the HANDLE ABEND block represents a link-level, so the table is scanned backwards for the highest active entry. If there are no active entries, the task is terminated abnormally.

If an active entry is found it is immediately inactivated.
HANDLE ABEND LABEL will cause a branch to the label.
HANDLE ABEND PROGRAM will cause an XCTL to the program.
If the abending program has received a COMMAREA, then that will be passed to the abend handler.

Note that PUSH, POP, HANDLE ABEND CANCEL/RESET only affect the current link-level.

Dumps...

A program check will always produce an ASRA SNAP dump.
ABEND without NODUMP, will always produce a dump using ABCODE.

No other dumps will be produced if an abend or condition is handled.

COMMAREA Management

a) RETURN COMMAREA

The program issuing the RETURN must be at link-level 1 (i.e. about to return to Z390KCP). If this is not the case then INVREQ will be raised. This condition cannot be IGNORED as it is assumed that no valid code follows a RETURN.

The RETURN macro sets TCTTECA (address) and TCTTECAL (length). When the next task is invoked TCTTECAL is used to refresh EIBCALEN.

When Z390KCP regains control after RETURN, the COMMAREA address and length are compared with the last RETURN COMMAREA (holding areas COMMADDR and COMMLLEN).

--- If both are the same, they are passed to the next transid.

--- If either is different, then a new area is GETMAINd, the COMMAREA is copied, and the old one FREEMAINd.
Temporary holding areas COMMSAVA and COMMSAVL are used during the FREEMAIN process.

b) LINK COMMAREA

The address is stored in the linkers DSA at DFHEICAP and the length in EIBCALEN. DFHEICAP is passed as a parameter.

Note: A LINK COMMAREA is never specifically FREEMAINd, it is always part of another storage area (DSA, Program, GETMAIN).

c) XCTL COMMAREA

The current COMMAREA address in the DSA (DFHEICAP) and length in EIBCALEN are compared with the XCTL COMMAREA.

--- If both are the same, the address is passed to the next program.

--- If either is different, then a new area is GETMAINd and the COMMAREA is copied.

The new COMMAREA address is held in the callers DSA (DFHEICAP) and is passed via a small GETMAINd area. This area address is stored at DFHEIPRM and its existence flagged by TCTTECND=X'FF'.

GETMAIN/FREEMAIN Management

ZCICSDRF.TXT

A chain of storage areas is anchored from TCTTESCC.
TCTTESCC has the address of the first GETMAINd area.

Eight bytes are added to each request, and they serve as a prefix...
4-byte address of next GETMAIN or 0, 4-byte total length.

The user is passed the address after the prefix.

FREEMAIN must have the same address as the GETMAIN passed otherwise an INVREQ condition is raised. FREEMAINS may occur in any order, the chain is just 'repaired' at that point.

At task end or task abend all remaining GETMAINS are FREEMAINd.

HANDLE AID Management

A HANDLE AID is owned by a program and is never passed to another program. The AID block (DFHADBLK, see above) is acquired on first use. PUSH and POP will acquire/release additional AID blocks.

At task end, task abend or an XCTL, all AID blocks are FREEMAINd.

HANDLE AID only works for conversational tasks.

HANDLE/IGNORE CONDITION Management

A HANDLE CONDITION is owned by a program and is never passed to another program. The CONDITION block (DFHHCBLK, see above) is acquired on first use. PUSH and POP will acquire/release additional CONDITION blocks.

Each 4-byte entry represents a condition, this may contain...

4X'00' -- The condition is not handled (default)
A(label) -- The condition will be handled at label
4X'FF' -- The condition should be ignored

The ERROR condition can be handled as a 'catch-all' for any type of condition that doesn't have a specific HANDLE CONDITION.
When both a condition and ERROR are set, only the condition label is used, not both.

At task end, task abend or an XCTL, all CONDITION blocks are FREEMAINd.

Temporary Storage Management

The queues are owned by Z390CICS, so all requests for TS services are sent by Z390/CICS tasks to the server.

There are two structures in the server:

The queue name chain...

A chain of all queue names.

The anchor of the chain is internal (TSNANCHR).

The DSECT for the name table is internal (TSNAMES).

A queue name is created by the first WRITEQ TS for the name and is chained on the end.

DELETEQ TS will delete all the data items and then delete the queue name and repair the chain.

The TS data chain...

A chain of all items added to the queue.

The anchor of the chain is in the queue name table (TSNITEM1).

The DSECT for the TS data chain prefix is internal (TSDPREFIX), the data follows the prefix.

ZCICSDRF.TXT

WRITEQ TS will add a new item to the chain end.
WRITEQ TS REWRITE will free the old item, create a new one,
and repair the chain.

DELETEQ TS will delete all the data items and then delete the
queue name and repair the chain.

File Control Management

The files are owned by Z390CICS, so all requests for FC services
are sent by Z390/CICS tasks to the server.

The FCT (DSECT DFHFCTDS) defines the status of each file.

Each FCTTE contains the ACB for that file.

See ZCICSVSM - Z390/CICS VSAM Guide to see how files are created and
defined to Z390/CICS.

a) File opening

When Z390CICS starts, all files defined as FILSTAT=OPENED are
opened.

Failure results in the status (CLOSED, DISABLED).

Files defined as (CLOSED, ENABLED) are opened when the first
request is received.

Failure results in the status (CLOSED, DISABLED).

b) Request processing

Any feedback codes are converted to RESP/RESP2 values and sent
back to the Client.

When a task ends or abends all VSWAs owned by the task are
released.

i) READ (ESDS)

A VSWA is acquired.

RPL OPTCD is set to (ADR) or (ADR, XRBA).

RPLARG is set to the address of (X)RBA.

Area of the maximum or fixed length is GETMAINd and RPLAREA
is set. GET issued and the data is sent to the program.

The VSWA is released.

ii) STARTBR (ESDS)

A check is made to ensure that a VSWA for the same REQID has
not already been acquired for this task (INVREQ/33).

A VSWA is acquired.

The REQID is set (default is zero).

RPL OPTCD is set to (ADR, SEQ) or (ADR, SEQ, XRBA).

RPLARG is set to the address of (X)RBA.

A POINT is issued.

The current XRBA is saved in the VSWA and returned to the
program.

iii) READNEXT (ESDS)

The VSWA created by the STARTBR is searched for, an error is
raised if not found (INVREQ/34).

RPL OPTCD is set to (ADR, SEQ, FWD) or (ADR, SEQ, FWD, XRBA).

RPLARG is set to the address of (X)RBA.

A check is made to see if the XRBA supplied differs from the
current XRBA, if it does then a POINT is issued. This allows
skip-sequential processing to occur.

Area of the maximum or fixed length is GETMAINd and RPLAREA
is set.

ZCICSDRF.TXT

The current XRBA is saved in the VSWA, a GET is issued, and the current (X)RBA and the data are sent to the program.

iii) READPREV (ESDS)

The VSWA created by the STARTBR is searched for, an error is raised if not found (INVREQ/41).

RPL OPTCD is set to (ADR, SEQ, BWD) or (ADR, SEQ, BWD, XRBA).

RPLARG is set to the address of (X)RBA.

A check is made to see if the XRBA supplied differs from the current XRBA, if it does then a POINT is issued. This allows skip-sequential processing to occur.

Area of the maximum or fixed length is GETMAINd and RPLAREA is set.

The current XRBA is saved in the VSWA, a GET is issued, and the current (X)RBA and the data are sent to the program.

iv) RESETBR (ESDS)

The VSWA created by the STARTBR is searched for, an error is raised if not found (INVREQ/36).

RPL OPTCD is set to (ADR, SEQ) or (ADR, SEQ, XRBA).

RPLARG is set to the address of (X)RBA.

A POINT is issued.

The current XRBA is saved in the VSWA and returned to the program.

v) ENDBR (ESDS)

The VSWA created by the STARTBR is searched for, an error is raised if not found (INVREQ/35).

The VSWA is released.

Event Tracing

Tracing is currently limited to those events that the server knows about. These events appear on the log as WTO messages.

The level of tracing is controlled by the INI parm TRACE_Z390CICS=

There is an intent to provide full application tracing, but each event would have to be sent to the server and may be too great an overhead in this environment.

Internal Abends (Z390CICS)

These are mostly caused by programming errors or situations I had not anticipated. Please report all of these to Don.

555	Unknown request sent to Server
666	CMDPROC failed
777	TCPIO OPEN/CLOSE Server failed
778	TCPIO RECEIVE failed
780	TCPIO SEND failed
790	VSAM feedback code was not expected

Trademarks

CICS is a registered trademark of International Business Corporation.

Author: Melvyn Maltz

Shipping Date: October 1st, 2007

Z390 version: V1.3.08

Z390/CICS version: V3

→