CICS - ERROR HANDLING

http://www.tutorialspoint.com/cics/cics_error_handling.htm

Copyright © tutorialspoint.com

There are many types of abends and errors which one can face while using CICS application. Errors can be because of both hardware of software issues. We will be discussing about errors and error handling in this module.

CICS Errors

Following are the CICS errors which can arise during execution of CICS applications:

- Some expected CICS errors arises when the conditions are not normal in CICS system. For example, if we are reading a particular record and record is not found then we get "Not Found" error. Mapfail is a similar error. Errors in this category are handled by explicit logic in the program.
- Logical errors arises because of some reasons like division by zero, illegal character in numeric field or transaction id error.
- Errors that are related to hardware or other system conditions are beyond the control of an application program. For example, getting input/output error while accessing a file.

Error Handling Commands

CICS provides several mechanisms to identify the errors and to handle them in our programs. Following are the commands which are used to handle the expected CICS errors:

- Handle condition
- Handle Abend
- Abend
- Ignore Condition
- Nohandle

We will discuss each of them in detail now.

Handle Condition

Handle condition is used to transfer the control of program to a paragraph or a procedure label. If the condition name specified in the exception block arises the particular para will be given control and then we can handle that condition. HANDLE CONDITION can handle only conditions related to CICS not the ordinary program Abends like as data exceptions. It can handle conditions that are related only to CICS. It can not handle the ordinary program Abends like as data exceptions. Following it the syntax of Handle Condition:

```
EXEC CICS HANDLE CONDITION
    CONDITION(Label)
    CONDITION(Label)
    ERROR(LABEL)
END-EXEC.
```

Example

Following is the example of Handle condition:

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.
PROCEDURE DIVISION.
EXEC CICS HANDLE CONDITION
    DUPKEY(X0000-DUPKEY-ERR-PARA)
    NOTFND(X000-NOT-FOUND-PARA)
    ERROR(X0000-GEN-ERR-PARA)
END-EXEC.
X0000-DUPKEY-ERR-PARA.
DISPALY 'Duplicate Key Found'.
X0000-NOT-FOUND-PARA.
DISPLAY 'Record Not Found'.
X0000-GEN-ERR-PARA.
DISPLAY 'General Error'.
```

Handle Abend

If a program abends due to some reasons like input-output error, etc, then it can be handled using Handle Abend CICS command. Following is the syntax of Handle Abend command:

```
 EXEC CICS HANDLE ABEND
    PROGRAM(name)
    LABEL(Label)
    CANCEL
    RESET
END-EXEC
```

Program name or label name is used to transfer the control to the program or paragraph if abend occurs. CANCEL is used to cancel previous HANDLE CONDITIONS. RESET is used to re activate the previously cancelled HANDLE ABEND.

Example

Following is the example of Handle Abend:

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.
PROCEDURE DIVISION.
EXEC CICS HANDLE ABEND
    LABEL (X0000-HANDLE-ABEND-PARA)
END-EXEC.
X0000-HANDLE-ABEND-PARA.
DISPLAY 'Program Abended'.
```

Abend

Abend command is use to terminate the task intentionally. Using Abend command we can set a user defined abend code. Following is the syntax of Abend command:

```
EXEC CICS ABEND
    ABCODE(name)
END-EXEC.
```

Example

Following is the example of Abend command. It will abend when the program reaches this paragraph with the user defined abend code, like in below example it will abend with abend code D100:

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.
PROCEDURE DIVISION.
EXEC CICS ABEND
    ABCODE(D100)
END-EXEC.
```

Ignore Condition

Ignore condition is used when we want no action should be taken if a particular abend or error happens which is mentioned inside the Ignore Condition. Following is the syntax of Ignore Condition:

```
EXEC CICS IGNORE CONDITION
    CONDITION(Label)
END-EXEC.
```

Example

Following is the example of Ignore Condition. It will not abend the program even if program throws length error as we have mentioned LENGERR inside Ignore condition.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.
PROCEDURE DIVISION.
EXEC CICS IGNORE CONDITION
    LENGERR
END-EXEC.
```

Nohandle

Nohandle can be specified for any CICS command . This will cause no action to be taken for any exceptional conditions that may occur during the execution of the CICS command. This command temporarily deactivates all the other handle conditions. If an exception arises during execution of the command the control will be transferred to the next statement after the Command. It can be used with Read, Write, Delete, etc. Following is the syntax of Nohandle:

```
EXEC CICS
    program statements
    NOHANDLE
END-EXEC.
```

Example

Following is the example of Nohandle command. We are using it with a Read statement. If Read statement fails, it will not abend the program.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.
PROCEDURE DIVISION.
EXEC CICS READ
    FILE('FILE1')
    INTO(WS-FILE-REC)
    RIDFLD(WS-STDID)
    NOHANDLE
END-EXEC.
```