Summer University

# Discover the CICS Catalog Manager

*The CICS TS V3.1 sample application*

Lab Version V1.00

Tuesday, 23 July, 2013

## *Overview*

With CICS TS V3.1 comes a sample application called CICS Catalog Manager. This is basically a catalog ordering system for office materials whose modular design makes it perfectly suitable for modernization and reuse. This lab will introduce the Catalog Manager and show you a little more detail about the program.

## *Scenario*

You are new to a company. This company has been using a 3270 catalog ordering system for many years. You are asked to modernize the application. You will first explore the program and its code and find out the program's interfaces.

## *Lab Requirements*

This lab assumes that the CICS Catalog Manager has been installed to CICS. Furthermore you will need WDz to explore the program contents.

## *Lab Steps*

**Part 1** Fehler! Ungültiger Eigenverweis auf Textmarke.

You will explore the application using its 3270 interface.

### Part 2 Create Filters to display Catalog Manager Contents

In this part WDz will be used to create a filter for suitable displaying the contents.

### Part 3 The Catalog Manager Program Interfaces

This part contains a summary of the catalog manager's functions and interfaces as reference for later labs.

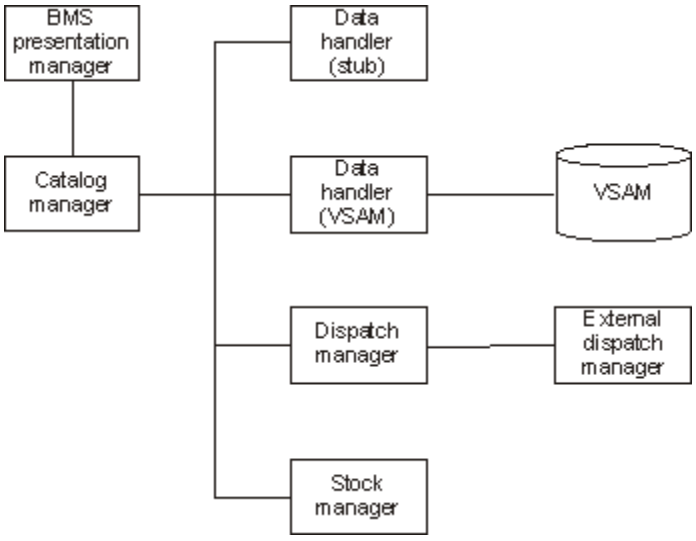
# Part 1: Familiarize yourself with the CICS catalog manager application

The CICS catalog example is a working COBOL application that is designed to illustrate best practice when connecting CICS applications to external clients and servers.

The example is constructed around a simple sales catalog and order processing application, in which the end user can perform these functions;

- List the items in a catalog.
- Inquire on individual items in the catalog.
- Order items from the catalog.

The base application has a 3270 user interface, but the modular structure, with well-defined interfaces between the components, makes it possible to add further components. In particular, the application comes with Web service support, which is designed to illustrate how you can extend an existing application into the Web services environment. The catalog data is stored in VSAM.



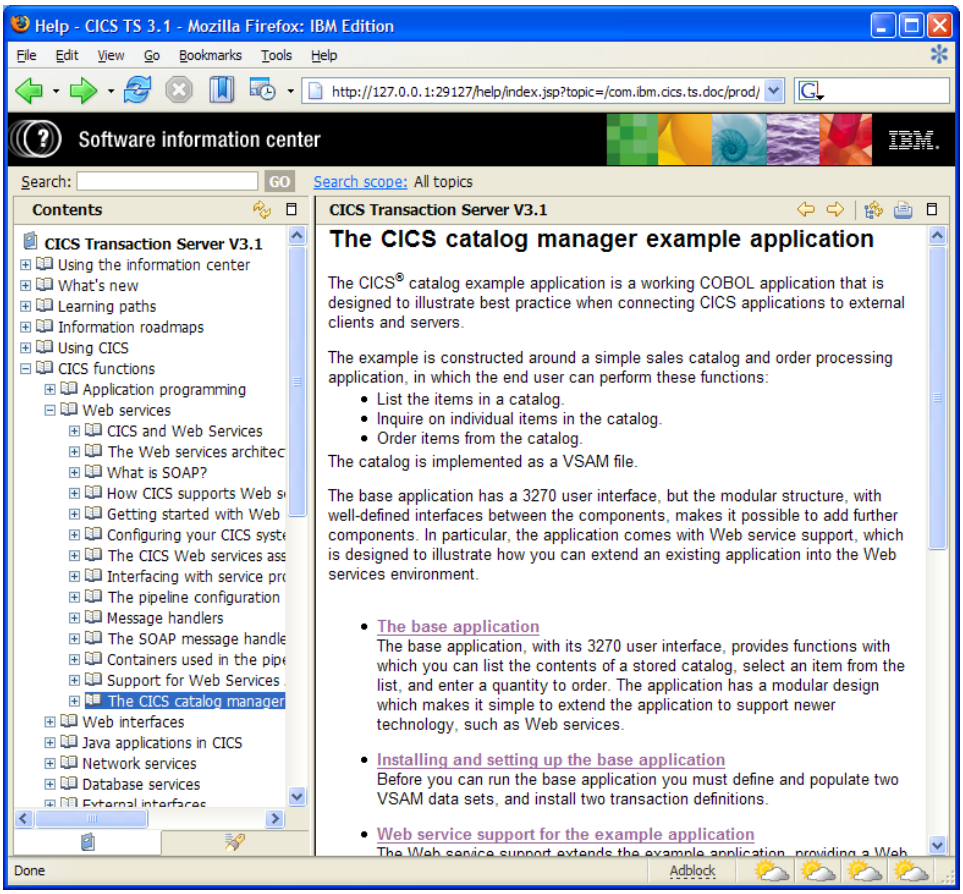

The example application has already been installed and the 3270 user interface configured ready for your use. The CSD group DFH$EXBS contains the application resource definitions.

## *Start the CICS Information Center*

It will be useful to have the CICS TS Information Center up and running so you can easily find out more about the catalog manager application or look up aspects of the Web services support in CICS during the workshop.

There are a number of ways to start the Information Center. The easiest is to use the direct online version available from http://www.ibm.com/software/htp/cics/tserver/v31/library/ . However, you may wish to have it installed either as part of an integrated development environment (IDE), or run it on a server in order to work disconnected or with no reliance on an internet connection.

Select “CICS Transaction Server V3.1” in the Contents pane, then click on the + symbol to expand “CICS functions”, then expand “Web services”, then select “The CICS catalog manager example application”.



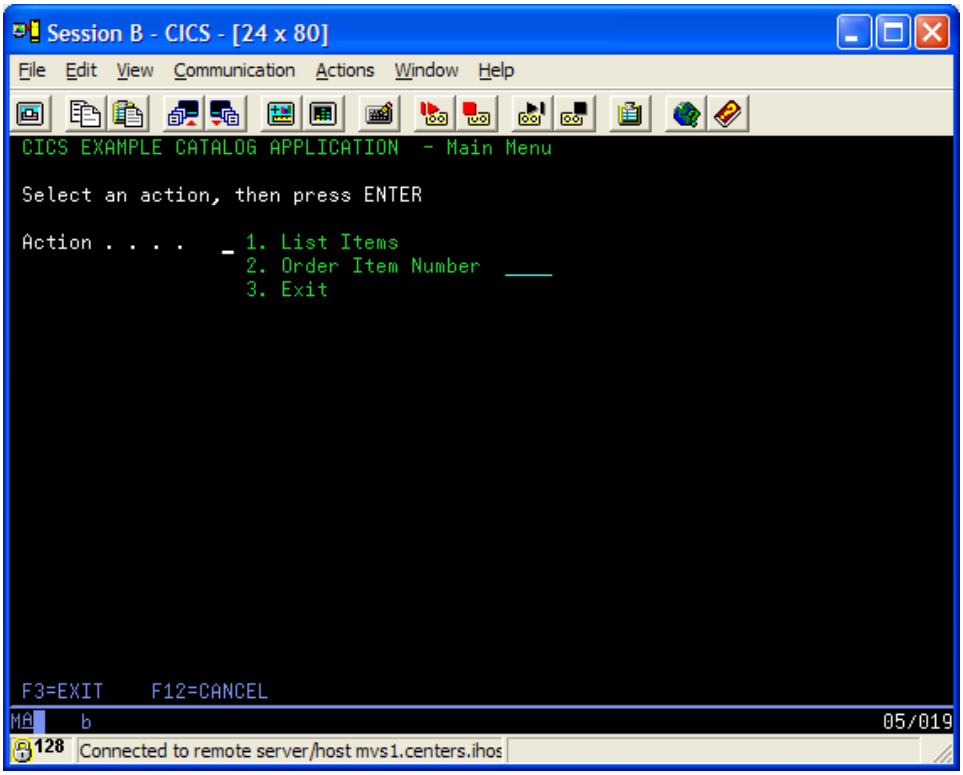
## *Running the example application with the 3270 interface*

Right click your Remote System Connection in your WDz workspace and select Host Connection Emulator Support.

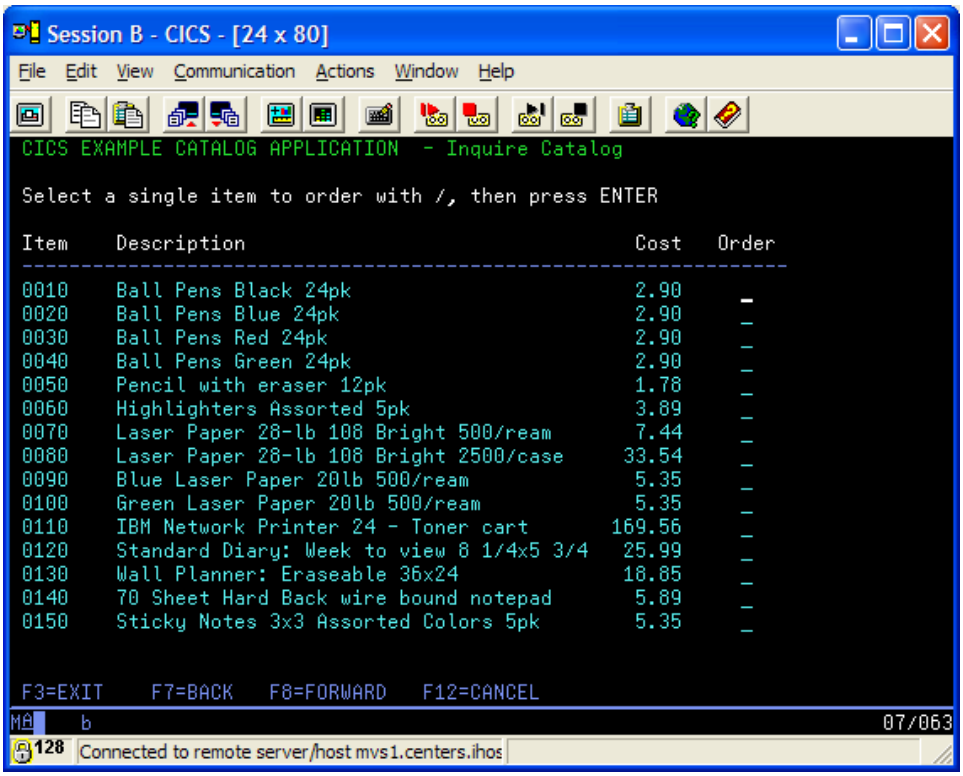
Start CICS TS V3.1

Type `[CLEAR]`

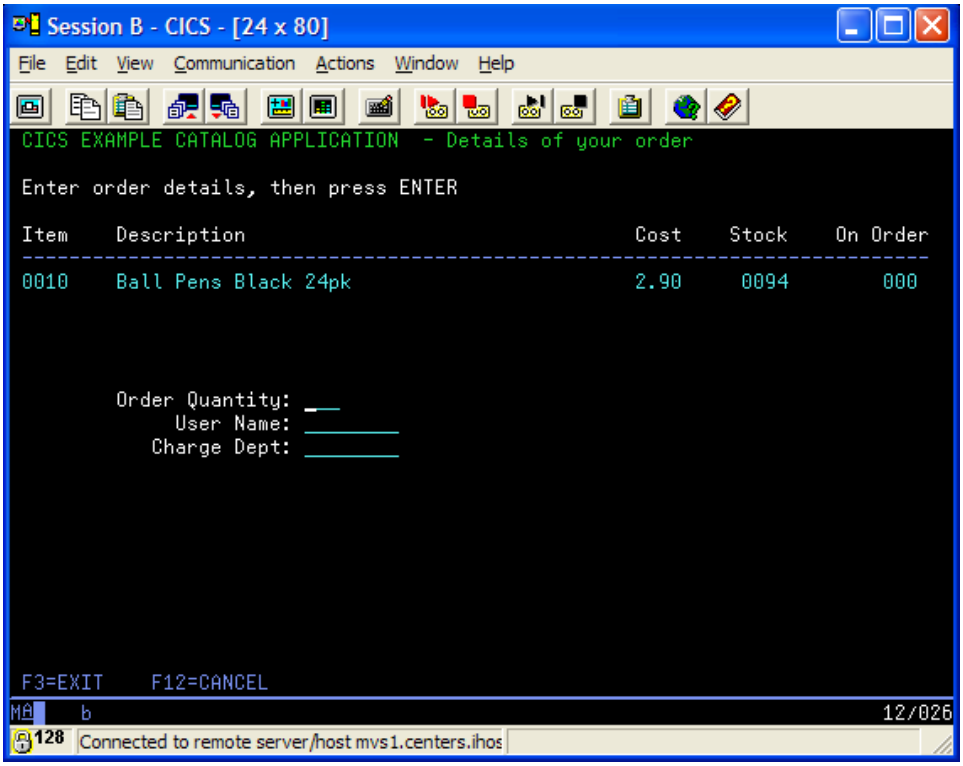
Type `EGUI [ENTER]` to start the example catalog application. The options on the main menu enable you to list the items in the catalog, order an item, or exit the application. You should see the following panel:



Type `1` `[ENTER]` in the Action field to select List Items. The application displays a list of items in the catalog. This function is referred as inquireCatalog.




Type `/` `[ENTER]` in the Order column next to item 0010 to order this item. The application displays details of the item to be ordered. The query of a single element is referred as InquireSingle.



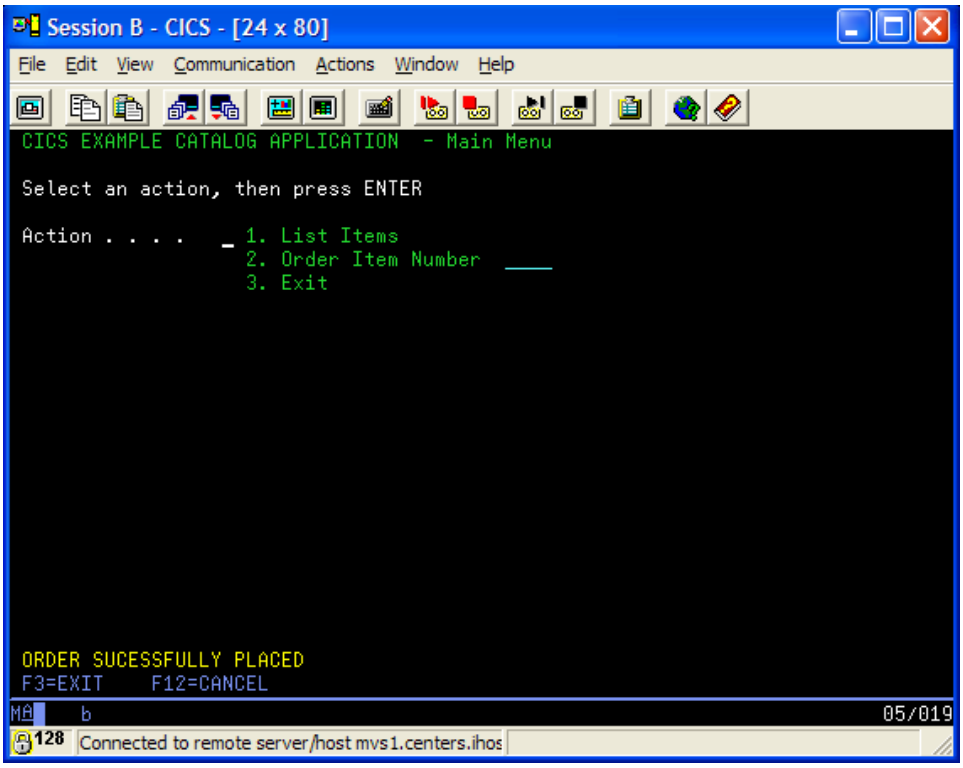

Providing there is sufficient stock to fulfill the order, enter the following information.

Type `1` in the Order Quantity field.

Type your `<TSO_UserID>` in the User Name field. In fact any 1 to 8-character string will be fine - the base application does not check the value that is entered.

Type `WORKSHOP [ENTER]` in the Charge Dept field. In fact any 1 to 8-character string will be fine – the base application does not check the value that is entered.

The Main Menu panel and the message “ORDER SUCCESSFULLY PLACED” is now displayed. This function is referred as placeOrder.




You could use the List Items function to check the stock levels were been updated.

Type `[F3] [CLEAR]` to end the application.

# Part 2: Create Filters to display Catalog Manager Contents

To be able to access the Catalog Manager contents on the file system you will now create a filter in your workspace and specify mappings according to the occurring file types.

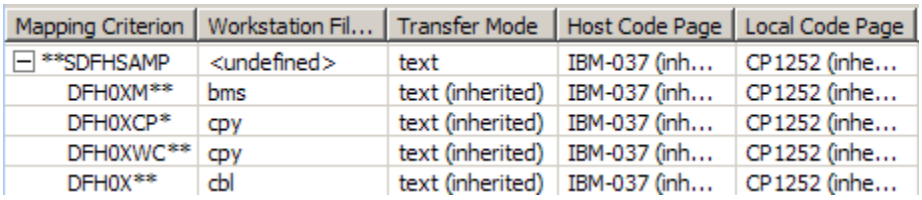
Right click MVS Files in your Remote System connection and select New → Filter. Create a MVS filter with the following properties

| | |
|---|---|
| Filter String | ☆CATMAN_LOCATION☆ |
| Filter name | catman |

You will find all resources there. Specify the following mappings to have the right file extensions. Use the z/OS File System Mapping View to define the following mappings

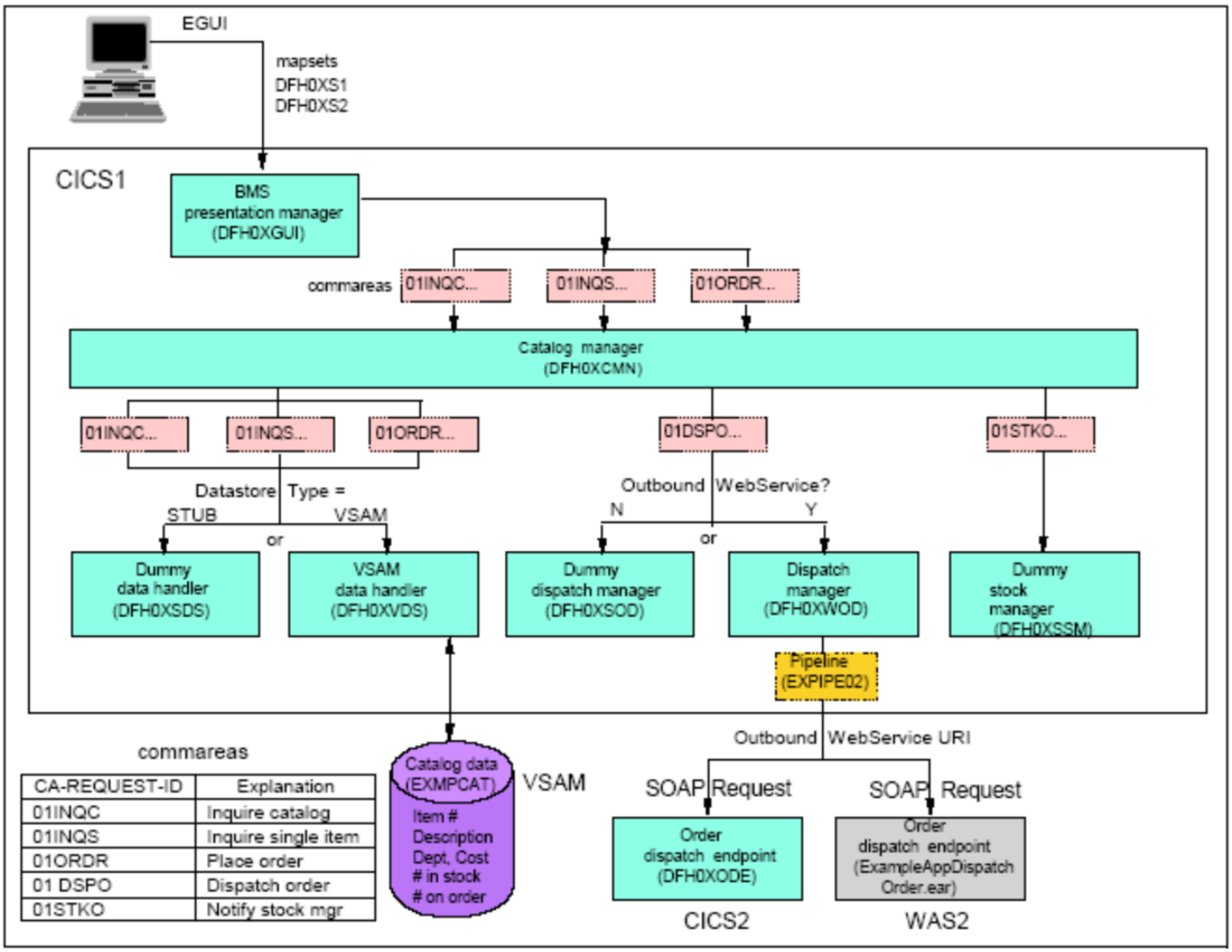
| Mapping Type | Mapping Criterion | Workstation File Extension |
|---|---|---|
| Data Set Mapping | **SDFHSAMP | Default (undefined) |
| Member Mapping | DFH0XM** | bms |
| Member Mapping | DFH0XCP** | cpy |
| Member Mapping | DFH0XWC** | cpy |
| Member Mapping | DFH0X** | cbl |

The filters are prioritized, so make sure to define the most common filter for cbl as the last one. Your filters should look like



| Mapping Criterion | Workstation Fil... | Transfer Mode | Host Code Page | Local Code Page |
|---|---|---|---|---|
| **SDFHSAMP | <undefined> | text | IBM-037 (inh... | CP1252 (inhe... |
| DFH0XM** | bms | text (inherited) | IBM-037 (inh... | CP1252 (inhe... |
| DFH0XCP* | cpy | text (inherited) | IBM-037 (inh... | CP1252 (inhe... |
| DFH0XWC** | cpy | text (inherited) | IBM-037 (inh... | CP1252 (inhe... |
| DFH0X** | cbl | text (inherited) | IBM-037 (inh... | CP1252 (inhe... |

Now expand your filter. The following picture will show the usage of some of the files



```
CONFIGURE CICS EXAMPLE CATALOG APPLICATION


           Datastore Type ==> VSAM               STUB!VSAM
     Outbound WebService? ==> YES                YES!NO
          Catalog Manager ==> DFH0XCMN
          Data Store Stub ==> DFH0XSDS
          Data Store VSAM ==> DFH0XVDS
       Order Dispatch Stub ==> DFH0XSOD
Order Dispatch WebService ==> DFH0XWOD
            Stock Manager ==> DFH0XSSM
           VSAM File Name ==> EXMPCAT
  Server Address and Port ==> myserver:99999
  Outbound WebService URI ==> http://139.18.4.35:03601/exampleApp/dispatch
                          ==> Order
                          ==>
                          ==>
                          ==>
```

The catalog manager design is modular, it consists of several program modules which are specified in a configuration file. You can change the configuration using the ECFG transaction in CICS:

The BMS presentation manager (DFH0XEGUI) is the 3270 front end you just used to run the application. From there the CICS Catalog Manager program (DFH0XCMN) is called using a distributed program link and submitting a COMMAREA. In this COMMAREA a request-id is included to specify the function that should be called (inquireSingle, InquireCatalog or placeOrder).

Depending on whether you specify STUG or VSAM  as dataset type the items will be requested from a stub program (DFH0XSDS) or from a VSAM dataset (via program DFH0XVDS calling the VSAM catalog EXMPCAT).

Depending on whether you wish to dispatch an order to an outbound web service or not the stub program DFH0XSOD or the program DFH0XWOD (sending a request to the Outbound WebService URI) will be called.

Open at least DFH0XCMN to become familiar with the program. Note that DFH0XCMN gets all the program names it has to call from the configuration file EXMPCONF.

# Part 3: The Catalog Manager Program Interfaces

For reference and later use we summarized the functions and interfaces of the catalog manager for you. You can also refer to the CICS Infocenter
http://publib.boulder.ibm.com/infocenter/cicsts/v3r1/topic/com.ibm.cics.ts31.doc/dfhxa/topics/dfhxa_t512.htm

## *INQUIRE SINGLE ITEM operation*

This operation returns a single catalog item specified by the caller.
The data structures are defined in the copybook DFH0XCP4.

| Input parameters | |
|---|---|
| CA-REQUEST-ID | A string that identifies the operation. For the INQUIRE SINGLE ITEM command, the string contains "01INQS" |
| CA-ITEM-REF-REQ | The reference number of the item to be returned. |
| Output parameters | |
| CA-RETURN-CODE | 0 if operation successful |
| CA-RESPONSE-MESSAGE | A human readable string, containing RETURNED ITEM: REF=item-reference' where item-reference is the reference number of the returned item. |
| CA-SINGLE-ITEM | An array containing in its first element the returned catalog item. |

## *INQUIRE CATALOG operation*

This operation returns a list of up to 15 catalog items, starting with the item specified by the caller. The data structures are defined in the copybook DFH0XCP3.

| Input parameters | |
|---|---|
| CA-REQUEST-ID | A string that identifies the operation. For the INQUIRE CATALOG command, the string contains "01INQC" |
| CA-LIST-START-REF | The reference number of the first item to be returned. |
| Output parameters | |
| CA-RETURN-CODE | 0 if operation successful |
| CA-RESPONSE-MESSAGE | A human readable string, containing "num ITEMS RETURNED" where num is the number of items returned. |
| CA-LAST-ITEM-REF | The reference number of the last item returned. |
| CA-ITEM-COUNT | The number of items returned. |
| CA-CAT-ITEM | An array containing the list of catalog items returned. The array has 15 elements; if fewer than 15 items are returned, the remaining array elements contain blanks. |

## *PLACE ORDER operation*

This operation places an order for a single item. If the required quantity is not available a message is returned to the user. If the order is successful, a call is made to the Stock Manager informing it what item has been ordered and the quantity ordered.
The data structures are defined in the copybook DFH0XCP5.

| Input parameters | |
|---|---|
| CA-REQUEST-ID | A string that identifies the operation. For the PLACE ORDER operation, the string contains '01ORDR' |
| CA-USERID | An 8-character user ID which the application uses for dispatch and billing. |
| CA-CHARGE-DEPT | An 8-character department ID which the application uses for dispatch and billing. |
| CA-ITEM-REF-NUMBER | The reference number of the item to be ordered. |
| CA-QUANTITY-REQ | The number of items required. |
| Output parameters | |
| CA-RETURN-CODE | 0 if operation successful |
| CA-RESPONSE-MESSAGE | A human readable string, containing 'ORDER SUCCESSFULLY PLACED'. |

## *NOTIFY STOCK MANAGER operation*

This operation takes details of the order that has been placed to decide if stock replenishment is necessary.The data structures are defined in the copybook DFH0XCP2.

| Input parameters | |
|---|---|
| CA-ORD-REQUEST-ID | A string that identifies the operation. For the NOTIFY STOCK MANAGER operation, the string contains '01STKO' |
| CA-STK-ITEM-REF-NUMBER | The reference number of the item to be ordered. |
| CA-STK-QUANTITY-REQ | The number of items required. |
| Output parameters | |
| CA-ORD-RETURN-CODE | 0 if operation successful |

## *DISPATCH STOCK operation*

This operation places a call to the stock dispatcher program, which in turn dispatches the order to the customer.

The data structures are defined in the copybook DFH0XCP6, 7 and 8.

| Input parameters | |
|---|---|
| CA-ORD-REQUEST-ID | A string that identifies the operation. For the DISPATCH ORDER operation, the string contains '01DSPO' |
| CA-ORD-USERID | An 8-character user ID which the application uses for dispatch and billing. |
| CA-ORD-CHARGE-DEPT | An 8-character department ID which the application uses for dispatch and billing. |
| CA-ORD-ITEM-REF-NUMBER | The reference number of the item to be ordered. |
| CA-ORD-QUANTITY-REQ | The number of items required. |
| Output parameters | |
| CA-ORD-RETURN-CODE | 0 if operation successful |

## *Return codes*

Each operation of the catalog manager can return a number of return codes.

| Type | Code | Explanation |
|---|---|---|
| General | 00 | Function completed without error |
| Catalog file | 20 | Item reference not found |
| | 21 | Error opening, reading, or ending browse of catalog file |
| | 22 | Error updating file |
| Configuration file | 50 | Error opening configuration file |
| | 51 | Data store type was neither STUB nor VSAM |
| | 52 | Outbound Web service switch was neither Y nor N |
| Remote Web service | 30 | The EXEC CICS INVOKE WEBSERVICE command returned an INVREQ condition |
| | 31 | The EXEC CICS INVOKE WEBSERVICE command returned an NOTFND condition |
| | 32 | The EXEC CICS INVOKE WEBSERVICE command returned a condition other than INVREQ or NOTFND |
| Application | 97 | Insufficient stock to complete order |
| | 98 | Order quantity was not a positive number |
| | 99 | DFH0XCMN received a COMMAREA in which the CA-REQUEST-ID field was not set to one of the following: 01INQC, 01INQS, or 01ORDR |