**Extending a BOD(Business Object Document) service to manage UserData with the Data Service layer**

We are creating a custom query template file to map an XPath expression and access profile to SQL. The template file performs the following tasks:

1. Defines an SQL template statement that fetches data from the new tables.
2. Associates a new access profile to the new SQL template statement.
3. Associates an XPath expression to the new SQL template statement.

The WebSphere Commerce BOD programming model separates processing into the business logic layer and the persistence layer. The business logic layer works with logical service data objects (SDOs), which are the Java representation of nouns. The persistence layer creates, reads, updates, or deletes business objects in the WebSphere Commerce database. This separation allows adding new data to affect only the persistence layer.

Step 1: **Customizing the WebSphere Commerce schema**

Run Following command on DB2 client to create tables

CREATE TABLE XWARRANTY (CATENTRY\_ID BIGINT NOT NULL, WARTERM INTEGER, WARTYPE VARCHAR(32), OPTCOUNTER SMALLINT, CONSTRAINT XWARRANTY\_PK PRIMARY KEY(CATENTRY\_ID), CONSTRAINT XWARRANTY\_FK FOREIGN KEY (CATENTRY\_ID) REFERENCES CATENTRY(CATENTRY\_ID) ON DELETE CASCADE);

CREATE TABLE XCAREINSTRUCTION (

CATENTRY\_ID BIGINT NOT NULL,

LANGUAGE\_ID INTEGER NOT NULL,

CAREINSTRUCTION VARCHAR(254),

OPTCOUNTER SMALLINT,

CONSTRAINT XCAREINST\_PK PRIMARY KEY (CATENTRY\_ID, LANGUAGE\_ID),

CONSTRAINT XCAREINST\_FK1 FOREIGN KEY (CATENTRY\_ID, LANGUAGE\_ID) REFERENCES CATENTDESC(CATENTRY\_ID, LANGUAGE\_ID) ON DELETE CASCADE,

CONSTRAINT XCAREINST\_FK2 FOREIGN KEY (CATENTRY\_ID) REFERENCES CATENTRY (CATENTRY\_ID) ON DELETE CASCADE

);

* The XWARRANTY table has a foreign key to the CATENTRY table. The key allows the Data Service Layer (DSL) to populate data from the XWARRANTY table in the user data element of the CatalogEntry noun.
* The XCAREINSTRUCTION table has a foreign key to the CATENTDESC table to allow DSL to populate data in the attribute element of the CatalogEntryDescription noun part.
* The XCAREINSTRUCTION table has a foreign key to the CATENTRY table to provide support to the DSL search function.

Populate the following data in the tables created

INSERT INTO XWARRANTY (CATENTRY\_ID, WARTERM, WARTYPE) VALUES (*10002*, 30, 'LIMITED');

INSERT INTO XWARRANTY (CATENTRY\_ID, WARTERM, WARTYPE) VALUES (*10004*, 45, 'COMPREHENSIVE');

INSERT INTO XWARRANTY (CATENTRY\_ID, WARTERM, WARTYPE) VALUES (*10017*, 60, 'LIMITED');

INSERT INTO XCAREINSTRUCTION (CATENTRY\_ID, LANGUAGE\_ID, CAREINSTRUCTION) VALUES (*10002*, -1, 'Never use an abrasive cleaner or material on any finished product');

INSERT INTO XCAREINSTRUCTION (CATENTRY\_ID, LANGUAGE\_ID, CAREINSTRUCTION) VALUES (*10004*, -1, 'Avoid soap and water');

INSERT INTO XCAREINSTRUCTION (CATENTRY\_ID, LANGUAGE\_ID, CAREINSTRUCTION) VALUES (*10017*, -1, 'Never use household cleaners');

2. Step2: **Customizing the physical layer to include the new information**

Customizing the physical layer by modifying the WebSphere Commerce schema, generating object-relational metadata, and generating physical service data objects (SDOs). To do so, we use a tool called the Data Service Layer wizard.

The Data Service Layer wizard is used to generate object-relational metadata and physical data objects for your schema customization. Physical SDOs are service data objects that represent tables in the WebSphere Commerce schema. Each data object type corresponds to a table definition in the schema, and each data object property corresponds to a table column or a reference to another DataObject type. These references correspond to the foreign key relationships between the database tables.

For each service module, there is object-relational metadata that contains the information to relate the physical DataObject to a database table. Custom object-relational metadata is stored in the component configuration extension directories and custom physical SDOs are stored inside the WebSphereCommerceServerExtensionsLogic project.

This wizard performs the following tasks:

1. Creates an extension configuration folder for the Catalog service module if one does not exist. The directory path is:*[WC\_eardir](https://www.ibm.com/support/knowledgecenter/SSZLC2_7.0.0/com.ibm.commerce.base.doc/misc/mabhelp.htm?lang=en-us)*\xml\config\com.ibm.commerce.catalog-ext.
2. Generates a custom object-relational metadata file that describes the new custom tables and relationships. In this tutorial, the metadata file describes the two new tables, XWARRANTY and XCAREINSTRUCTION, along with the three new relationships between tables.

* XWARRANTY and CATENTRY
* XCAREINSTRUCTION and CATENTRY
* XCAREINSTRUCTION and CATENTDESC

This file location is [WC\_eardir](https://www.ibm.com/support/knowledgecenter/SSZLC2_7.0.0/com.ibm.commerce.base.doc/misc/mabhelp.htm?lang=en-us)\xml\config\com.ibm.commerce.catalog-ext\wc-object-relational-metadata.xml.

1. Generates an SDO Java class and places it in the WebSphereCommerceServerExtensionsLogic project for:

* Each new custom table (XWARRANTY and XCAREINSTRUCTION).
* Each modified WebSphere Commerce table (CATENTRY and CATENTDESC were modified by adding new relationships to the custom tables).

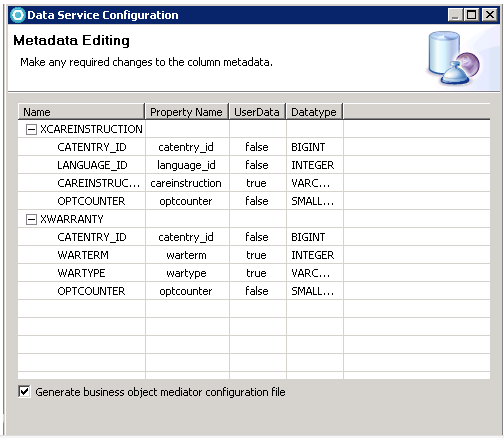
Physical SDOs are service data objects that represent tables in WebSphere Commerce

1. Creates a utility Java class to return the physical SDO root class (and its package) for the service module. This root class ensures that all WebSphere Commerce physical SDOs for the Service Module, and any additional physical SDOs for the customization, are available at run time.
2. Creates an extension service module configuration file that instructs WebSphere Commerce to use the newly created catalog physical SDO class in [*WC\_eardir*](https://www.ibm.com/support/knowledgecenter/SSZLC2_7.0.0/com.ibm.commerce.base.doc/misc/mabhelp.htm?lang=en-us)\xml\config\com.ibm.commerce.catalog-ext\wc-component.xml.
3. Creates an extension business object mediator configuration file. This file configures the business object mediator to include data from the XWARRANTY and XCAREINSTRUCTION tables in the user data of a CatalogEntry noun. This file location is[*WC\_eardir*](https://www.ibm.com/support/knowledgecenter/SSZLC2_7.0.0/com.ibm.commerce.base.doc/misc/mabhelp.htm?lang=en-us)\xml\config\com.ibm.commerce.catalog-ext\wc-business-object-mediator.xml.

**Procedure**

To begin customizing the physical layer:

1. If your development environment uses an Apache Derby database, ensure that there is no existing connection to it. For example, ensure that the WebSphere Commerce server is stopped.
2. Select **File** > **New** > **Other** > **WebSphere Commerce** > **Data Service Layer**.
3. Click **Next**.
4. Select **Extend a default WebSphere Commerce service module**
5. Click **Next**.
6. Enter the following information:
   * Service module: Select **com.ibm.commerce.catalog**
   * Extension class prefix: MyCompany
   * Extension Java package name: com.mycompany.commerce.catalog
7. Click **Next.**
8. Select the XCAREINSTRUCTION and XWARRANTY tables.
9. Click **Next**.
10. Under **UserData**, ensure the Warterm, Wartype, and Careinstruction database columns are set to **True**. Set the other columns to**False** if they are set as True by default.



Step 3: **Adding query templates to include the new information**

The query template file consists of the following components:

1. A symbol definition section that defines the tables our query template uses (CATENTRY, CATENTDESC, XWARRANTY, XCAREINSTRUCTION.
2. An XPath to SQL statement that maps the XPath key and access profile to a specific template SQL query.
3. A new access profile, MyCompany\_All that is used along with the XPath key to identify the SQL template query.

The default queries to fetch the data before updating the CatalogEntry noun, and CatalogEntryDescription noun part must be changed to include the XWARRANTY and XCAREINSTRUCTION tables.

* The default SELECT queries for updating the CatalogEntry nouns and its parts is located inside the following file:
  + [*WC\_eardir*](https://www.ibm.com/support/knowledgecenter/SSZLC2_7.0.0/com.ibm.commerce.base.doc/misc/mabhelp.htm?lang=en-us)\xml\config\com.ibm.commerce.catalog\wc-query-CatalogEntry-update.tpl
  + [*WC\_eardir*](https://www.ibm.com/support/knowledgecenter/SSZLC2_7.0.0/com.ibm.commerce.base.doc/misc/mabhelp.htm?lang=en-us)\xml\config\com.ibm.commerce.catalog-fep\wc-query-CatalogEntry-update.tpl
* The default SELECT query used to update the CatalogEntry noun is identified by the IBM\_CatalogEntryUpdate access profile. The default SELECT query to update the CatalogEntryDescription noun part is identified by the IBM\_CatalogEntryDescription\_Update access profile.

1. Create a custom Get query template file:
2. In Enterprise Explorer, expand **WC** > **xml** > **config** and right click the com.ibm.commerce.catalog-ext folder. (If the com.ibm.commerce.catalog-ext folder is not visible, select the WC\config folder and select **File** > **Refresh**.)
3. Select **New** > **File**.
4. Name the file: wc-query-MyCompanyCatalogEntry-get.tpl

The name is important – it must begin with wc-query- and end with the suffix .tpl

1. Click **Finish**.
2. Copy the following

BEGIN\_SYMBOL\_DEFINITIONS

<!-- WebSphere Commerce tables -->

COLS:CATENTRY=CATENTRY:\*

COLS:CATENTDESC=CATENTDESC:\*

<!-- MyCompany extension tables -->

COLS:XWARRANTY=XWARRANTY:\*

COLS:XCAREINSTRUCTION=XCAREINSTRUCTION:\*

END\_SYMBOL\_DEFINITIONS

BEGIN\_XPATH\_TO\_SQL\_STATEMENT

name=/CatalogEntry[CatalogEntryIdentifier[(UniqueID=)]]+MyCompany\_All

base\_table=CATENTRY

sql=

SELECT

CATENTRY.$COLS:CATENTRY$,

CATENTDESC.$COLS:CATENTDESC$,

XWARRANTY.$COLS:XWARRANTY$,

XCAREINSTRUCTION.$COLS:XCAREINSTRUCTION$

FROM

CATENTRY

LEFT OUTER JOIN XWARRANTY ON (CATENTRY.CATENTRY\_ID = XWARRANTY.CATENTRY\_ID)

LEFT OUTER JOIN CATENTDESC ON (CATENTDESC.CATENTRY\_ID = CATENTRY.CATENTRY\_ID AND CATENTDESC.LANGUAGE\_ID in ($CONTROL:LANGUAGES$))

LEFT OUTER JOIN XCAREINSTRUCTION ON (CATENTRY.CATENTRY\_ID = XCAREINSTRUCTION.CATENTRY\_ID AND XCAREINSTRUCTION.LANGUAGE\_ID = CATENTDESC.LANGUAGE\_ID)

WHERE

CATENTRY.CATENTRY\_ID IN (?UniqueID?) AND

CATENTRY.MARKFORDELETE = 0

END\_XPATH\_TO\_SQL\_STATEMENT

The two subsections of this example query template are:

**SYMBOL\_DEFINITIONS**

In this subsection you define symbols for the columns used in your SQL. If your physical schema changes, you can adjust the symbols without rewriting all of your SQL.

**XPATH\_TO\_SQL\_STATEMENT**

Maps an XPath expression directly to an SQL statement. In the example above, an XPath statement was defined, named, and mapped to a base table and an SQL statement.

1. Save the file.
2. Create a custom update query file:
3. Right-click the com.ibm.commerce.catalog-ext folder.
4. Select **New** > **File**.
5. Name the file: wc-query-MyCompanyCatalogEntry-update.tpl.
6. Click **Finish**.
7. Copy and paste the following query template into the file.

BEGIN\_SYMBOL\_DEFINITIONS

<!-- WebSphere Commerce tables -->

COLS:CATENTRY=CATENTRY:\*

COLS:CATENTDESC=CATENTDESC:\*

<!-- MyCompany extension tables -->

COLS:XWARRANTY=XWARRANTY:\*

COLS:XCAREINSTRUCTION=XCAREINSTRUCTION:\*

END\_SYMBOL\_DEFINITIONS

BEGIN\_ASSOCIATION\_SQL\_STATEMENT

name=MyCompanyWarrantyCatalogEntry

base\_table=CATENTRY

additional\_entity\_objects=true

sql=

SELECT

CATENTRY.$COLS:CATENTRY$,

XWARRANTY.$COLS:XWARRANTY$

FROM

CATENTRY

JOIN STORECENT ON STORECENT.CATENTRY\_ID=CATENTRY.CATENTRY\_ID

LEFT OUTER JOIN XWARRANTY ON (CATENTRY.CATENTRY\_ID = XWARRANTY.CATENTRY\_ID)

WHERE

CATENTRY.CATENTRY\_ID IN ($ENTITY\_PKS$)

AND STORECENT.STOREENT\_ID IN ($STOREPATH:catalog$)

AND CATENTRY.MARKFORDELETE=0

END\_ASSOCIATION\_SQL\_STATEMENT

BEGIN\_XPATH\_TO\_SQL\_STATEMENT

name=/CatalogEntry[CatalogEntryIdentifier[(UniqueID=)]]/Description+MyCompany\_CatalogEntryDescription\_Update

base\_table=CATENTDESC

sql=

SELECT

CATENTDESC.$COLS:CATENTDESC$,

XCAREINSTRUCTION.$COLS:XCAREINSTRUCTION$

FROM

CATENTDESC

LEFT OUTER JOIN XCAREINSTRUCTION ON (CATENTDESC.CATENTRY\_ID = XCAREINSTRUCTION.CATENTRY\_ID AND XCAREINSTRUCTION.LANGUAGE\_ID = CATENTDESC.LANGUAGE\_ID)

WHERE

CATENTDESC.CATENTRY\_ID IN (?UniqueID?)

END\_XPATH\_TO\_SQL\_STATEMENT

BEGIN\_PROFILE

name=MyCompany\_CatalogEntry\_Update

extends = IBM\_Admin\_CatalogEntryUpdate

BEGIN\_ENTITY

associated\_sql\_statement=MyCompanyWarrantyCatalogEntry

END\_ENTITY

END\_PROFILE

1. Save the file.
2. Update the wc-business-object-mediator.xml to instruct the Catalog service module to use the newly defined access profiles. This profile causes the Catalog service module to use the new queries, which include the custom tables, instead of the default queries provided by WebSphere Commerce.
3. Open WC\xml\config\com.ibm.commerce.catalog-ext\wc-business-object-mediator.xml.
4. Select the source view and find the following element:

<\_config:object logicalType="com.ibm.commerce.catalog.facade.datatypes.CatalogEntryType"

physicalType="com.mycompany.commerce.catalog.facade.server.entity.datatypes.MyCompanyCatalogEntry">

1. Copy and paste the following mediation configuration after the line you found in step 3b:

<\_config:mediator interfaceName="com.ibm.commerce.foundation.server.services.dataaccess.bom.mediator.ChangeBusinessObjectMediator"

className="com.ibm.commerce.catalog.facade.server.services.dataaccess.bom.mediator.ChangeCatalogEntryMediator"

updateAccessProfile="MyCompany\_CatalogEntry\_Update">

<\_config:part-mediator interfaceName="com.ibm.commerce.foundation.server.services.dataaccess.bom.mediator.ChangeBusinessObjectPartMediator">

<\_config:part-mediator-implementation className="com.ibm.commerce.catalog.facade.server.services.dataaccess.bom.mediator.ChangeCatalogEntryBasePartMediator" updateAccessProfile="MyCompany\_CatalogEntry\_Update"/>

<\_config:part-mediator-implementation className="com.ibm.commerce.catalog.facade.server.services.dataaccess.bom.mediator.ChangeCatalogEntryDescriptionMediator" updateAccessProfile="MyCompany\_CatalogEntryDescription\_Update"/>

</\_config:part-mediator>

</\_config:mediator>

1. Save the file.

Step 5: **Creating an access control policy to secure the new information**

The previous step created an access profile, MyCompany\_All. By default, only the users with a site administrator role have access to this new data. In this lesson, you update the Catalog service access control policy to state that all users have access to view this data.

The new policy defines a new action for the MyCompany\_All access profile and adds the new action to the CatalogEntry all users group. The access profiles for Change, Process, and Sync are only run after the access control check on the Change, Process, or Sync action.

**Procedure**

1. Create the following file: [*WCDE\_installdir*](https://www.ibm.com/support/knowledgecenter/SSZLC2_7.0.0/com.ibm.commerce.base.doc/misc/mabhelp.htm?lang=en-us)\xml\policies\xml\MyCompanyCatalogAccessControlPolicies.xml
2. Copy and paste the following access control policy XML into this file:

<Policies>

<Action Name="GetCatalogEntry.MyCompany\_All" CommandName="GetCatalogEntry.MyCompany\_All"/>

<ActionGroup Name="Catalog-CatalogEntry-AllUsers-AccessProfileActionGroup" OwnerID="RootOrganization">

<ActionGroupAction Name="GetCatalogEntry.MyCompany\_All"/>

</ActionGroup>

</Policies>

1. Run the **acpload** command to load the access control policies:
   1. Open a command prompt and go to *[WCDE\_installdir](https://www.ibm.com/support/knowledgecenter/SSZLC2_7.0.0/com.ibm.commerce.base.doc/misc/mabhelp.htm?lang=en-us)*\bin
   2. Run the acpload command with the following form:

acpload db\_host\_name db\_user db\_password inputXMLFile (NON-NLS)

eg acpload WCSDB db2admin Admin@123 MyCompanyCatalogAccessControlPolicies.xml db2admin

* 1. Check whether MyCompanyCatalogAccessControlPolicies\_idres.xml and MyCompanyCatalogAccessControlPolicies\_xmltrans.xml files are created in the directory MyCompanyCatalogAccessControlPolicies.xml is present. Also check for logs if there is any error.