

How to Access the CAF Entity Service Life Cycle Methods (Create, Read, Update, and Delete) from Web Dynpro

Applies to:

SAP NetWeaver 2004s (SP7, SP8) - Web Dynpro for Java and Composite Application Framework (CAF)

[Note: The delete operation will not work in SP10.]

Summary

This document explains how to access CAF Entity Service System define Method from Webdynpro Application. In CAF layer we write business logic. Some times Entity Service methods do not fulfill our business requirement in that case we crate our own custom define Application Service methods to write our own business logic. Create; read, update and delete are basic life cycle method of Entity service. We can't directly use Entity service **delete** and **update** method from web Dynpro because these two methods are not exposed to Web Dynpro. We have to access these two methods via application service where as you can directly use **create** and **read** method from Web Dynpro with out going through application service.

First we will create a Composite Application with Entity Service and Application Service methods and then we will create a WebDynpro project using CAF Web Dynpro model and also the presentation logic inside the Web Dynpro Project.

Author: Mithu Ghosh

Company: IBM India Private Limited

Created on: 30th Sep 2007

Author Bio



Mithu Ghosh is an application developer working in IBM India Private Limited on SAP NetWeaver Technology (CAF, WebDynpro for JAVA and GP)

Table of Contents

Preconditions	3
The Solutions with Implementation.....	3
Create a CAF Development Component Project.....	3
Create the Web Dynpro Application using CAF DC	6
Output Screen slots	18
Disclaimer and Liability Notice.....	21

Preconditions

The following are the preconditions that we need.

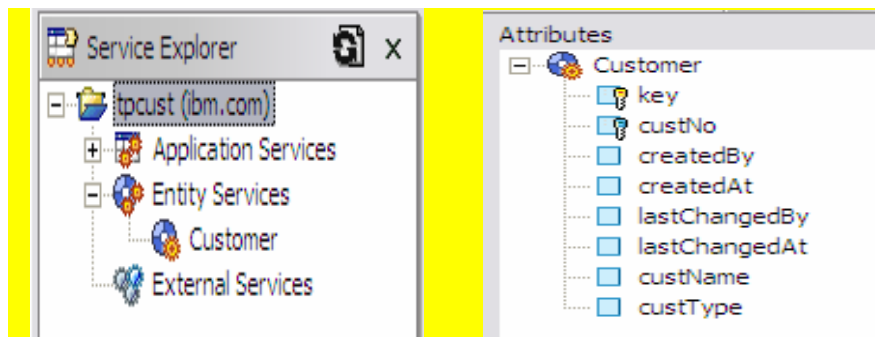
- Basic Concepts of Java programming. [[Java Tutorials](#)]
- Basic knowledge of Composite Application Framework. [[My First Composite Application](#)]
- An overview of Web Dynpro Programming [[Using a Composite Application via a Web Dynpro UI](#)]
- Before Deploying our Webdynpro Application we should check whether the configuration settings for the J2EE server and for the SDM server are entered correctly in the SAP NetWeaver Developer Studio or not.

To check the server settings, we have to choose the menu path *Window => Preferences=> SAP J2EE Engine*

The Solutions with Implementation

Create a CAF Development Component Project

Create a caf project. Create an Entity Service (**Customer**) and create some attribute (**custNo** as key, **custName**, **custType** of type `com.sap.caf.core.shortText`)



and operations (**getAllAvailableCustomer** with no parameter and **getCustomerByNumber** with parameter **custNo**).

Operation name	Visibility	Return type	Description
create	public	Customer	
read	public	Customer	
update	public	void	
delete	public	void	
findByMultipleParameters	public	*	
readByCustomKeys	public	Customer	
getAllAvailableCustomer	public	Collection	getAllAvailableCustomer
getCustomerByNumber	public	Collection	getCustomerByNumber

Input, Output Parameters and Exceptions

- Operation Parameters
 - Input
 - custNoFilter(com.sap.caf.rt.bol.util.QueryFilter)
 - Output
 - Result(java.util.List)
 - Fault
 - CAFFindException(null)

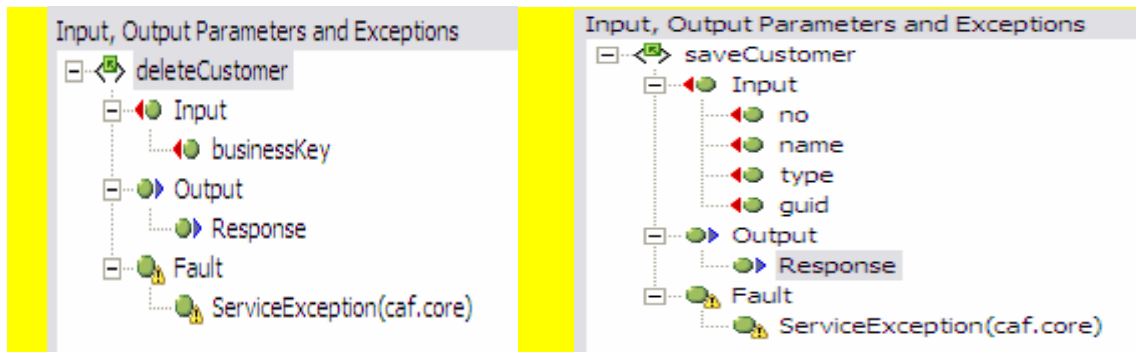
As we can't directly use Entity service method **delete** and **update** from web dynpro we have to create two application service methods to access these Entity service methods.

Now Create an Application Service (**CustomerApp**), in dependencies add the Entity Service (**Customer**) and add two custom operations (**deleteCustomer** and **saveCustomer**).

Name	Description	Transaction type	Permission	Operation type
deleteCustomer	deleteCustomer	Mandatory	Disabled	CUSTOM
saveCustomer	saveCustomer	Mandatory	Disabled	CUSTOM

In **deleteCustomer** method adds input parameter (**businessKey**) of type **com.sap.caf.core.id** and adds output parameter (**Response**) of type **com.sap.caf.core.shortText**.

In **saveCustomer** method adds input parameter (**no**, **name**, **type** of type **com.sap.caf.core.shortText** and **guid** of type **com.sap.caf.core.id**) and add output parameter (**Response**) of type **Customer** with **collection type** none.



Now write the following custom code in side the implementation of **deleteCustomer method**

```

@@custom code start - deleteCustomer(java.lang.String)
retValue = null;
CustomerServiceLocal custLocObj = this.getCustomerService();
Customer custM = custLocObj.read(businessKey); //read the existing
record
custLocObj.delete(custM); //delete the existing

```

and write the following custom code inside the implementation of **saveCustomer method**

```

@@custom code start - saveCustomer(java.lang.String, java.lang.String,
java.lang.String, java.lang.String)

retValue = null;
CustomerServiceLocal custservloc = this.getCustomerService();
Customer custM = custservloc.read(guid); //read the existing record
custM.setCustNo(no);
custM.setCustName(name);
custM.setCustType(type);

```

Now Create a *Web Dynpro Model* of the caf service and add the model into the caf webdynpro public part of DC Metadata.

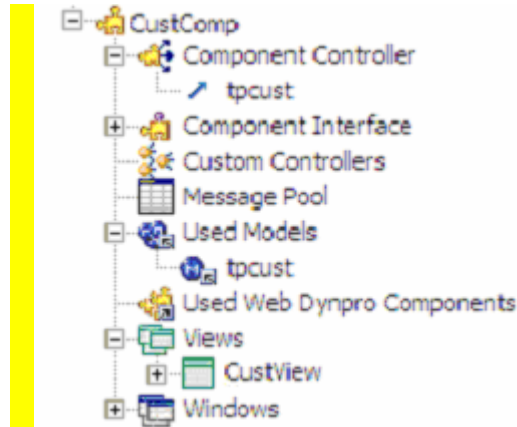
In the *Service Explorer* view, select the caf project **Generate Project Code. Build the Development Component;** choose **Development Component=>Build**. Choose **OK**. Check for errors and warnings in view **General User Output**.

Deploy the Component; choose **Development Component=> Deploy**.

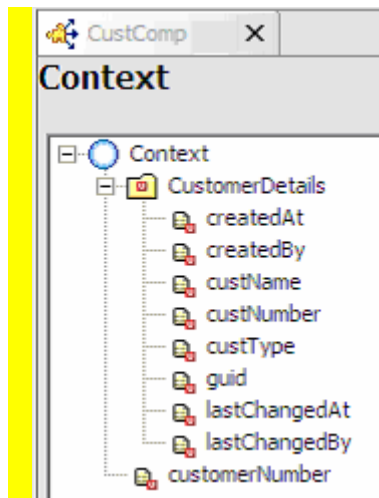
Add some records in CAF Entity Service (tpcust) from Service Browser.

Create the Web Dynpro Application using CAF DC

Create a Web Dynpro project adds the caf webdynpro model and the necessary DC in the *used DC* part of web Dynpro project. Create a Web Dynpro Component (**CustComp**) and add the caf model (**tpcust**) in the *used models* part.



Now In component controller (**CustComp**) context create a value node (**CustomerDetails**) with some value attribute (**createdAt** type timestamp, **createdBy** string, **custName** string, **custNumber** type string, **custType** type string, **guid** type string, **lastChangedAt** type timestamp, **lastChangedBy** type string) and a value attribute **customerNumber** of type string.

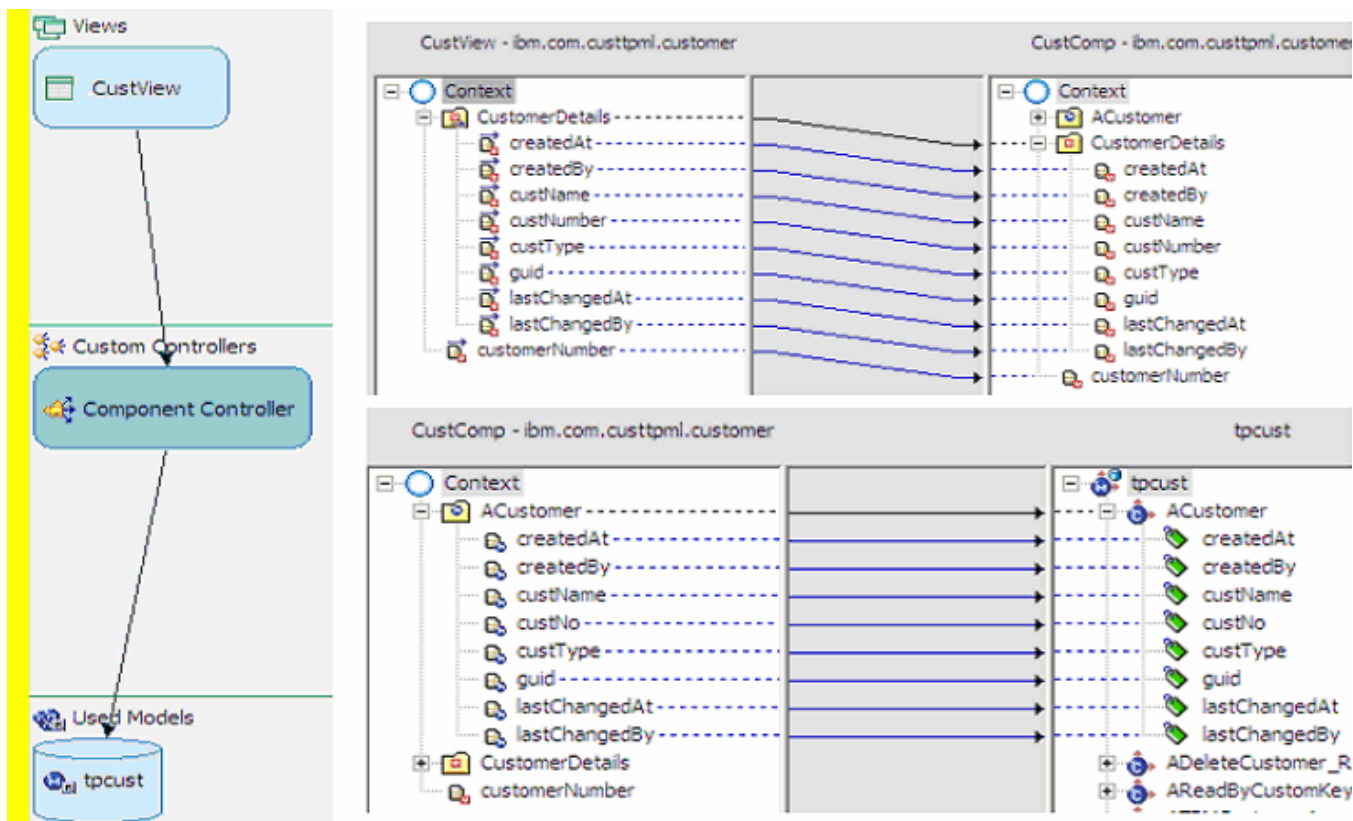


Create the following methods (**getAllCustomerDetails**, **getCustomerDetailByNumber** with parameter **cnum** of type **string**, **createCustomerDetails**, **saveCustomerDetails** and **deleteCustomerDetails**)

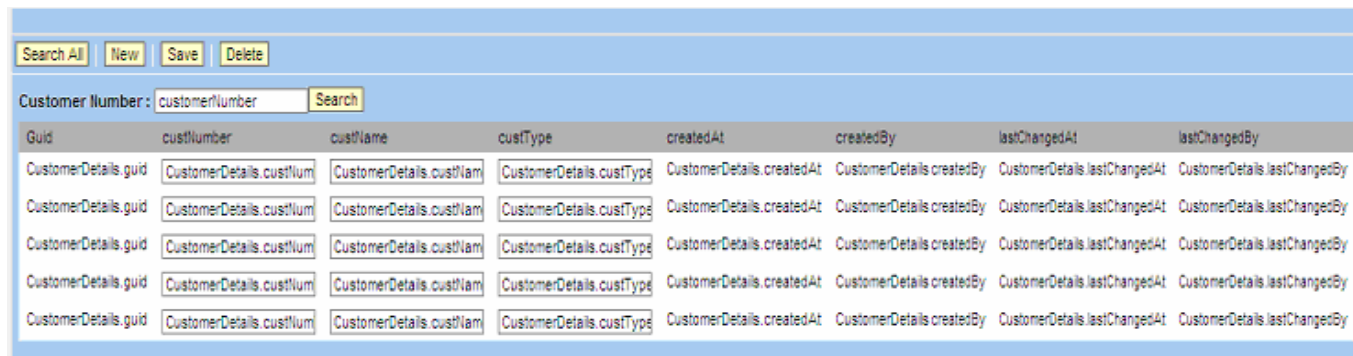
Methods
Displays the methods of the controller.

Type	Name	Return type
Method	createCustomerDetails	void
Method	deleteCustomerDetails	void
Method	getAllCustomerDetails	void
Method	getCustomerDetailByNumber	void
Method	saveCustomerDetails	void

In the diagram view do the mapping between *component controller* and *used model (caf model)*, *views* and *component controller*.



Design the layout of view (**CustView**).



Now create the following actions (**GoSearchAllCustomer**, **GoSearchCustomer**, **createNew**, **GoSave** and **GoDelete**).

Actions
Displays the actions of the controller

Name	W	Event handler	Text
GoDelete	<input type="checkbox"/>	onActionGoDelete	DeleteCustomer
GoSave	<input type="checkbox"/>	onActionGoSave	SaveCustomerDetails
GoSearchAllCustomer	<input type="checkbox"/>	onActionGoSearchAllCustomer	SeachAllCustomers
GoSearchCustomer	<input type="checkbox"/>	onActionGoSearchCustomer	SearchCustomer
createNew	<input checked="" type="checkbox"/>	onActioncreateNew	createNew

and in onAction Event property of **SerachAll** button associate the action **GoSearchAllCustomer**, with **Search** button associate **GoSearchCustomer**, with **New** button associate **createNew**, with **Save** button associate **GoSave** and with **Delete** button associate **GoDelete**.

Provide the following implementation with in the methods of View (**CustView**).

i> Inside **onActionGoSearchAllCustomer** method.

```
wdContext.currentContextElement().setCustomerNumber( " " );  
wdThis.wdGetCustCompController().getAllCustomerDetails();
```

ii> Inside **onActionGoSearchCustomer** method.

```
wdThis.wdGetCustCompController().getCustomerDetailByNumber(wdContext.currentContextElement().getCustomerNumber());
```

ii> Inside **onActioncreateNew** method.

```
wdThis.wdGetCustCompController().createCustomerDetails();
```

iii> Inside **onActionGoSave** method.

```
wdThis.wdGetCustCompController().saveCustomerDetails();
```

iv> Inside **onActionGoDelete** method.

```
wdThis.wdGetCustCompController().deleteCustomerDetails();
```

Provide the following implementation with in the methods of Component Controller (**CustComp**).

i> Inside **getAllCustomerDetails** method.

```

wdContext.nodeCustomerDetails().invalidate();
try {
    //retrive all records that are available in caf model using Entity
service proxy class
    IAspect aspect_list =
        CustomerServiceProxy.getAllAvailableCustomer();
wdContext.nodeACustomer().bind(aspect_list);
//bind the records in the model node.
IPublicCustComp.ICustomerDetailsElement custElm = null;
//populating the value node(CustomerDetails) by model node(ACustomer)
data
for (int i = 0; i < wdContext.nodeACustomer().size(); i++) {
    custElm = wdContext.createCustomerDetailsElement();
    custElm.setCreatedAt(
        (Timestamp) wdContext.nodeACustomer().getElementAt(
            i).getAttributeValue(
                "createdAt"));
    custElm.setCreatedBy(
        (String) wdContext.nodeACustomer().getElementAt(
            i).getAttributeValue(
                "createdBy"));
    custElm.setCustName(
        (String) wdContext.nodeACustomer().getElementAt(
            i).getAttributeValue(
                "custName"));
    custElm.setCustNumber(
        (String) wdContext.nodeACustomer().getElementAt(
            i).getAttributeValue(
                "custNo"));
    custElm.setCustType(
        (String) wdContext.nodeACustomer().getElementAt(
            i).getAttributeValue(
                "custType"));
    custElm.setLastChangedAt(
        (Timestamp) wdContext.nodeACustomer().getElementAt(
            i).getAttributeValue(
                "lastChangedAt"));
}

```

ii> Inside `getCustomerDetailByNumber(java.lang.String cnum)` method.

```

wdContext.nodeCustomerDetails().invalidate();

try {
    IAspect aspect_list =
        CustomerServiceProxy.getCustomerByNumber(cnum);
    //retrive the record by customer number using Entity service proxy
class
    wdContext.nodeACustomer().bind(aspect_list);
    //bind the record in the model node.
    IPublicCustComp.ICustomerDetailsElement custElm = null;
    //populating the value node(CustomerDetails) by model node(ACustomer)
data
    for (int i = 0; i < wdContext.nodeACustomer().size(); i++) {
        custElm = wdContext.createCustomerDetailsElement();
        custElm.setCreatedAt(
            (Timestamp) wdContext.nodeACustomer().getElementAt(
                i).getAttributeValue(
                    "createdAt"));
        custElm.setCreatedBy(
            (String) wdContext.nodeACustomer().getElementAt(
                i).getAttributeValue(
                    "createdBy"));
        custElm.setCustName(
            (String) wdContext.nodeACustomer().getElementAt(
                i).getAttributeValue(
                    "custName"));
        custElm.setCustNumber(
            (String) wdContext.nodeACustomer().getElementAt(
                i).getAttributeValue(
                    "custNo"));
        custElm.setCustType(
            (String) wdContext.nodeACustomer().getElementAt(
                i).getAttributeValue(
                    "custType"));
        custElm.setLastChangedAt(
            (Timestamp) wdContext.nodeACustomer().getElementAt(
                i).getAttributeValue(

```

iii> Inside **createCustomerDetails** method.

```
//create a new element
        wdContext.nodeCustomerDetails().addElement(
```

iv> Inside **saveCustomerDetails** method. // Create new records in caf local database as well as to update existing records

```
IWDMessageManager msg = wdComponentAPI.getMessageManager();

wdContext.nodeACustomer().invalidate();
for (int i = 0; i < wdContext.nodeCustomerDetails().size(); i++) {
    String cnum =
        wdContext
            .nodeCustomerDetails()
            .getCustomerDetailsElementAt(i)
            .getCustNumber();

    String cname =
        wdContext
            .nodeCustomerDetails()
            .getCustomerDetailsElementAt(i)
            .getCustName();

    String ctype =
        wdContext
            .nodeCustomerDetails()
            .getCustomerDetailsElementAt(i)
            .getCustType();

    String guid =
        wdContext
            .nodeCustomerDetails()
            .getCustomerDetailsElementAt(i)
            .getGuid();

    if (guid == null) //for new record
    {
        try {
            ACustomer custM = CustomerServiceProxy.create();
            //create a new dummy record in caf model with local guid
            custM.setCustNo(cnum);
            custM.setCustName(cname);
            custM.setCustType(ctype);
```

```
wdContext
    .nodeCustomerDetails()
    .getCustomerDetailsElementAt(
        i).setCustNumber(
            (String) aspect_lst crt.getAspectRow(
                0).getAttributeValue(
                    "custNo"));

wdContext
    .nodeCustomerDetails()
    .getCustomerDetailsElementAt(
        i).setCustType(
            (String) aspect_lst crt.getAspectRow(
                0).getAttributeValue(
                    "custType"));

wdContext
    .nodeCustomerDetails()
    .getCustomerDetailsElementAt(
        i).setCreatedAt(
            (Timestamp) aspect_lst crt.getAspectRow(
                0).getAttributeValue(
                    "createdAt"));

wdContext
    .nodeCustomerDetails()
    .getCustomerDetailsElementAt(
        i).setCreatedBy(
            (String) aspect_lst crt.getAspectRow(
                0).getAttributeValue(
                    "createdBy"));

wdContext
    .nodeCustomerDetails()
    .getCustomerDetailsElementAt(
        i).setGuid(
            (String) aspect_lst crt.getAspectRow(
                0).getAttributeValue(
                    "key"));

wdContext
    .nodeCustomerDetails()
    .getCustomerDetailsElementAt(
        i).setLastChangedAt(
            (Timestamp) aspect_lst crt.getAspectRow(
```

```

else //for existing record
{
    try {
        ACustomer custM = CustomerServiceProxy.read(guid);
        if ((cname.equals(custM.getCustName())
            && ctype.equals(custM.getCustType())
            && cnum.equals(custM.getCustNo())) {
            continue;
        } else {
            IAspect aspect_lst_upd =
                CustomerAppProxy.saveCustomer(
                    cnum,
                    ctype,
                    guid,
                    cname);

//update the existing record using Application service proxy class
//display the updated record
wdContext
    .nodeCustomerDetails()
    .getCustomerDetailsElementAt(
        i).setCustName(
        (String) aspect_lst_upd.getAspectRow(
            0).getAttributeValue(
                "custName"));

wdContext
    .nodeCustomerDetails()
    .getCustomerDetailsElementAt(
        i).setCustNumber(
        (String) aspect_lst_upd.getAspectRow(
            0).getAttributeValue(
                "custNo"));

wdContext
    .nodeCustomerDetails()
    .getCustomerDetailsElementAt(
        i).setCustType(
        (String) aspect_lst_upd.getAspectRow(
            0).getAttributeValue(

```

```

        wdContext
            .nodeCustomerDetails()
            .getCustomerDetailsElementAt(
                i).setGuid(
                    (String) aspect_lst_upd.getAspectRow(
                        0).getAttributeValue(
                            "key"));
    wdContext
        .nodeCustomerDetails()
        .getCustomerDetailsElementAt(
            i).setLastChangedAt(
                (Timestamp) aspect_lst_upd.getAspectRow(
                    0).getAttributeValue(
                        "lastChangedAt"));
    wdContext
        .nodeCustomerDetails()
        .getCustomerDetailsElementAt(
            i).setLastChangedBy(
                (String) aspect_lst_upd.getAspectRow(
                    0).getAttributeValue(
                        "lastChangedBy"));

    aspect_lst_upd.clear();

```

v> Inside `deleteCustomerDetails` method.

```

for (int i = 0; i < wdContext.nodeCustomerDetails().size(); i++) {
    try {
        if ((!wdContext.nodeCustomerDetails().isMultiSelected(i))
            && (wdContext.nodeCustomerDetails().getLeadSelection()
                != i)) {
            continue;
        } else {
            String guidID =
                wdContext
                    .nodeCustomerDetails()
                    .getCustomerDetailsElementAt(i)
                    .getGuid()
                    .toString();

```



```
// delete the particular customer details based on the guid using Application
service proxy class

        IAspect asp1 = CustomerAppProxy.deleteCustomer(guidID);
        IAspectRow iasprow = asp1.getAspectRow(0);
        String str_val =
            iasprow.getAttributeAsString("value").toString();
        wdComponentAPI.getMessageManager().reportSuccess(
            "CUSTOMER NUMBER : " + cno + str_val);
    }
} catch (Exception e) {
    wdComponentAPI.getMessageManager().reportException(
        "NO RECORD TO DELETE",
        - - -
```

Between `//@@begin` others and `//@@end`

```
IServiceFacade serviceFacadeProdConf =
    CAFServiceFactory.getServiceFacade(tncustDefinition class);
```

Right import the necessary packages click in the coding window using the alternate mouse button and then choose *Source -> Organize Imports*.

Create an application (**CustomerApp**) using existing component (**CustComp**)



Build the Component Select your Web Dynpro project; choose **Development Component=>Build**

Choose **OK**. Check for errors and warnings in view **General User Output**.

Deploy the Component, choose **Development Component=> Deploy**. If Successful

Run the **CustomerApp** Application.

Output Screen slots

	custNumber	custName	custType	createdAt	createdBy	lastChangedAt	lastChangedBy	Guid

Create New Customer: -

	custNumber	custName	custType	createdAt	createdBy	lastChangedAt	lastChangedBy	Guid
	101	cust1	type1					
	102	cust2	type2					
	103	cust3	type3					

Save CustomerDetails: -

	custNumber	custName	custType	createdAt	createdBy	lastChangedAt	lastChangedBy	Guid
	101	cust1	type1	2007-05-14T16:44:15.187	Guest	2007-05-14T16:44:15.203	Guest	400f2632-020c-11dc-9253-00145e3676ba
	102	cust2	type2	2007-05-14T16:44:48.656	Guest	2007-05-14T16:44:48.671	Guest	54021d02-020c-11dc-cb86-00145e3676ba
	103	cust3	type3	2007-05-14T16:44:48.718	Guest	2007-05-14T16:44:48.734	Guest	540b92e0-020c-11dc-c79e-00145e3676ba

Update CustomerDetails: -

Search All | New | Save | Delete

Customer Number : Search

	custNumber	custName	custType	createdAt	createdBy	lastChangedAt	lastChangedBy	Guid
	101	cust1	type1	2007-05-14T16:44:15.187	Guest	2007-05-14T16:44:15.203	Guest	400f2632-020c-11dc-9253-00145e3676ba
	102	cust22	type22	2007-05-14T16:44:48.656	Guest	2007-05-14T16:50:13.500	Guest	54021d02-020c-11dc-cb86-00145e3676ba
	103	cust33	type33	2007-05-14T16:44:48.718	Guest	2007-05-14T16:50:13.531	Guest	540b92e0-020c-11dc-c79e-00145e3676ba
	104	cust4	type4	2007-05-14T16:49:33.750	Guest	2007-05-14T16:49:33.765	Guest	fd00160-020c-11dc-9964-00145e3676ba

SearchAll CustomerDetails:-

Search All | New | Save | Delete

Customer Number : Search

	custNumber	custName	custType	createdAt	createdBy	lastChangedAt	lastChangedBy	Guid
	101	cust1	type1	2007-05-14T16:44:15.187	Guest	2007-05-14T16:44:15.203	Guest	400f2632-020c-11dc-9253-00145e3676ba
	102	cust22	type22	2007-05-14T16:44:48.656	Guest	2007-05-14T16:50:13.500	Guest	54021d02-020c-11dc-cb86-00145e3676ba
	103	cust33	type33	2007-05-14T16:44:48.718	Guest	2007-05-14T16:50:13.531	Guest	540b92e0-020c-11dc-c79e-00145e3676ba
	104	cust4	type4	2007-05-14T16:49:33.750	Guest	2007-05-14T16:49:33.765	Guest	fd00160-020c-11dc-9964-00145e3676ba

Delete CustomerDetails: -

Search All | New | Save | Delete

Customer Number : Search

	custNumber	custName	custType	createdAt	createdBy	lastChangedAt	lastChangedBy	Guid
	102	cust22	type22	2007-05-14T16:44:48.656	Guest	2007-05-14T16:50:13.500	Guest	54021d02-020c-11dc-cb86-00145e3676ba
	103	cust33	type33	2007-05-14T16:44:48.718	Guest	2007-05-14T16:50:13.531	Guest	540b92e0-020c-11dc-c79e-00145e3676ba

Customer Number : 101 Successfully Deleted.
Customer Number : 104 Successfully Deleted.

Search By Customer Number: -

Search All | New | Save | Delete

Customer Number : 103 Search

	custNumber	custName	custType	createdAt	createdBy	lastChangedAt	lastChangedBy	Guid
	103	cust33	type33	2007-05-14T16:44:48.718	Guest	2007-05-14T16:50:13.531	Guest	540b92e0-020c-11dc-c79e-00145e3676ba

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.