IBM Case Manager:
External Data Integration Lab

Table of Contents

1.0 External Data Integration Lab Overview

1.1 Lab Objective

- To familiarize participants with the External Data Integration Framework.
- To give participants an understanding of how data from external sources can be integrated into a case management solution using an External Data Service.

1.2 Lab Description

In this lab, you will integrate a sample auto warranty claims case management solution with an external data service. A sample external data service will be used to populate case data fields such as Customer Information, Vehicle Information and Parts description for a warranty replacement part.

1.3 Lab Scenario – Auto Warranty Claims Management Solution

You are the business analyst for Automotive Warranty Corp. You are responsible for building a solution to manage warranty claims submitted by customers. A claim is initiated when a dealer submits the claim information. The solution must integrate with the Customer Relationship Management (CRM) system where the customer data is stored, as well as the Vehicle Information Database and the Parts Database. You are asked to enhance the solution, to pull data from these external data sources into the case management solution.

To facilitate these integration scenarios with external data sources, Case Manager provides an external data service (EDS) REST protocol. The EDS service implemented on the REST protocol acts as the communication bridge between Case Manager and the external data source. To populate the case properties with this external data, you must implement an external data service that adheres to this REST protocol.

1.3.1 Document Types

A document type WarrantyClaim has been created which initiates the case creation. The case type has been configured to use this document as the initiating document for the case. By selecting the Map Properties option, we can map the document properties to the case properties. The CustomerID, Vehicle Identification Number (VIN) and PartsID are mapped from the document type to the case type, when a new case is created.

1.3.2 Case Types

AutoWarrantyClaim: This is the case type used to manage the auto warranty claims process. In this lab, when creating a new case, we will enter the CustomerID, VIN and PartsID and the registered external data service will be queried to pull in the customer information, vehicle information and the parts information.

2.0 Developing the External Data Service

.

Use the external data service only for retrieving data from an external source. For example, when a case worker creates or modifies a case, Case Manager Client saves the data that was received from the external data source in Content Engine. If the case worker modifies this data, Case Manager Client does not update the corresponding data in the external data source.

The procedure to implement the external data service is as follows:

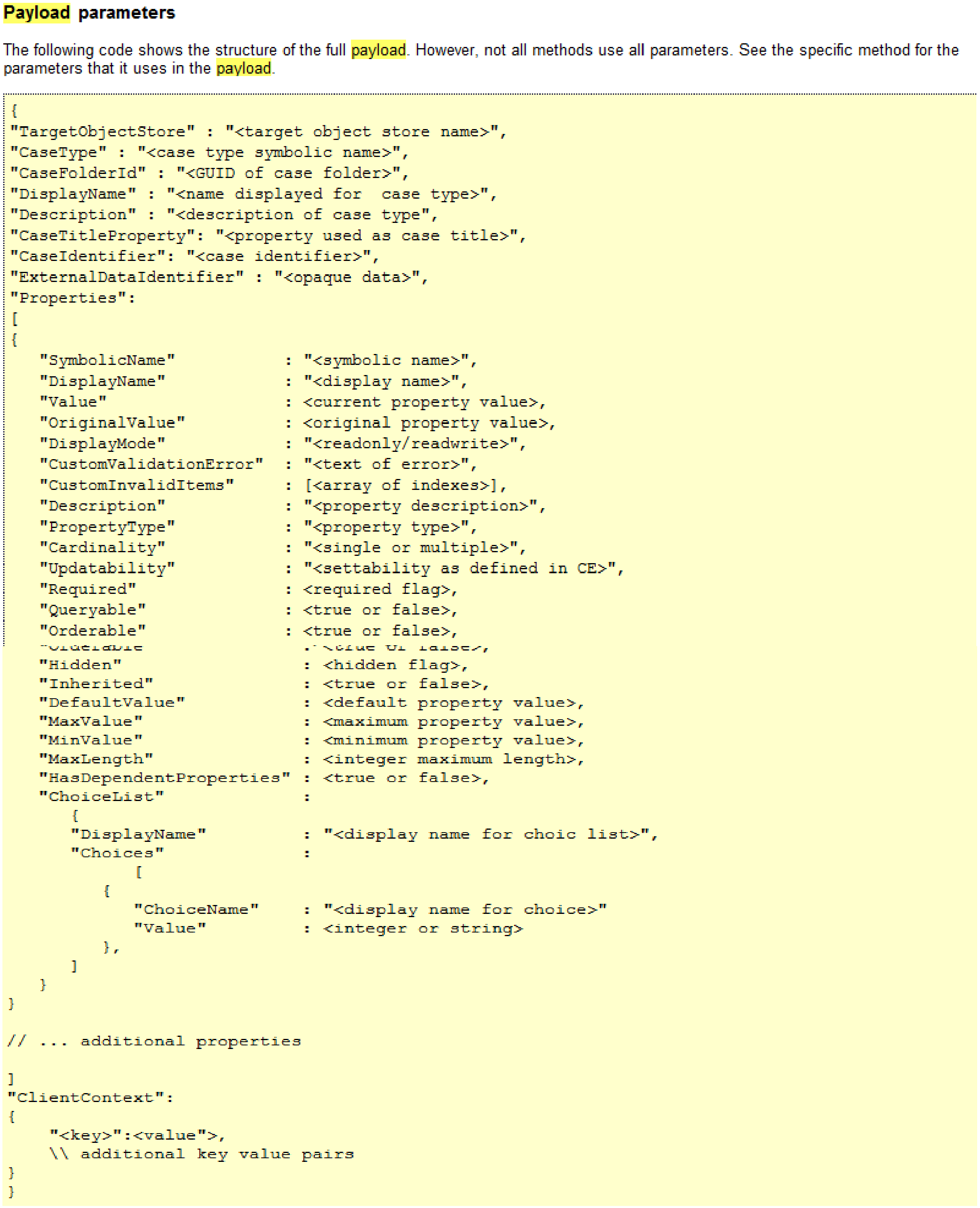
- Implement the HTTP POST method for the case type. This method returns the case properties that are to be populated in the case from the external data source.
- If the external data service needs to retrieve property attributes for case properties that it manages, then the service can use the Content Engine Java API or CMIS REST API to get those property attributes.
- If the external data service needs to authenticate users, it must participate in the single sign-on authentication configuration as do other Case Manager components.

In this lab example, the sample EDS service implements the HTTP POST method for the AutoClaimWarranty case type. When a case worker creates a new case or opens an existing case from the Case Manager Client, the Case Manager REST API automatically makes an HTTP POST call to the AutoClaimWarranty case type resource in the EDS. The POST method passes the request payload, which contains the current value for each case property. The current value can be one of the following:

- The default value that is specified in Case Builder for that case property
- The value persisted in Content Engine for the case property
- The working value entered by the case worker for that property

JSON Payload

The following is the generic payload structure that can be used with an external data service. In this lab you will use a supplied payload for the HTTP POST method.



Payload parameters

The following code shows the structure of the full payload. However, not all methods use all parameters. See the specific method for the parameters that it uses in the payload.

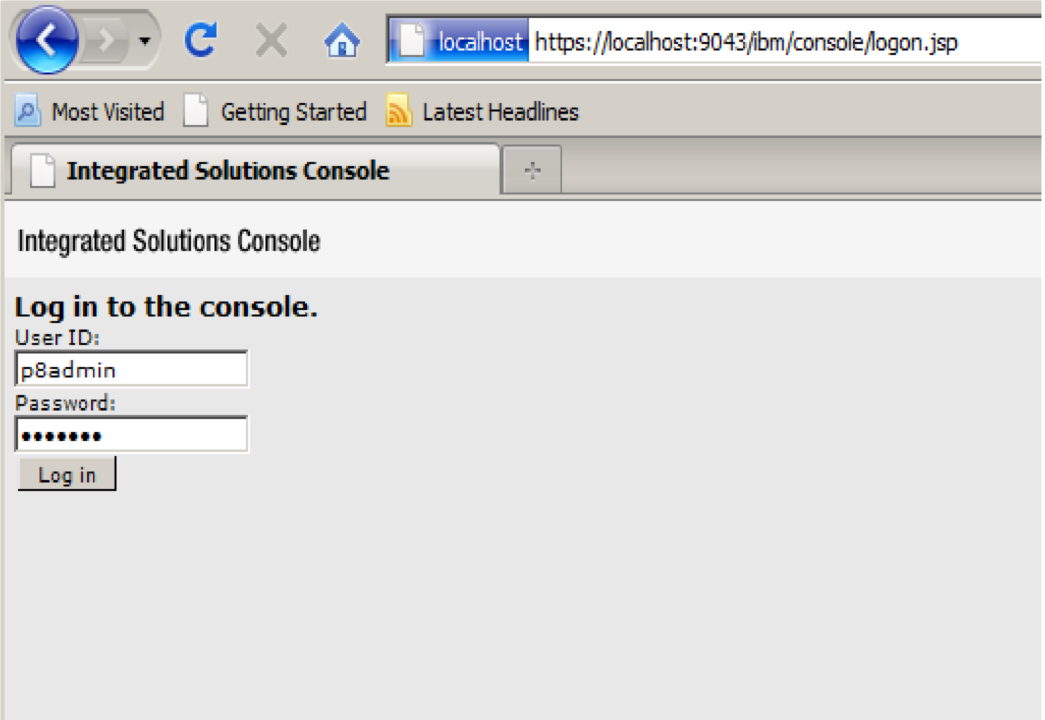
```
{
"TargetObjectStore" : "<target object store name>",
"CaseType" : "<case type symbolic name>",
"CaseFolderId" : "<GUID of case folder>",
"DisplayName" : "<name displayed for  case type>",
"Description" : "<description of case type",
"CaseTitleProperty": "<property used as case title>",
"CaseIdentifier": "<case identifier>",
"ExternalDataIdentifier" : "<opaque data>",
"Properties":
[
{
   "SymbolicName"            : "<symbolic name>",
   "DisplayName"             : "<display name>",
   "Value"                   : <current property value>,
   "OriginalValue"           : <original property value>,
   "DisplayMode"             : "<readonly/readwrite>",
   "CustomValidationError"   : "<text of error>",
   "CustomInvalidItems"      : [<array of indexes>],
   "Description"             : "<property description>",
   "PropertyType"            : "<property type>",
   "Cardinality"             : "<single or multiple>",
   "Updatability"            : "<settability as defined in CE>",
   "Required"                : <required flag>,
   "Queryable"               : <true or false>,
   "Orderable"               : <true or false>,
   "Hidden"                    : <hidden flag>,
   "Inherited"                 : <true or false>,
   "DefaultValue"              : <default property value>,
   "MaxValue"                  : <maximum property value>,
   "MinValue"                  : <minimum property value>,
   "MaxLength"                 : <integer maximum length>,
   "HasDependentProperties"    : <true or false>,
   "ChoiceList"                :
      {
      "DisplayName"            : "<display name for choic list>",
      "Choices"                :
            [
         {
            "ChoiceName"       : "<display name for choice>"
            "Value"            : <integer or string>
         },
      ]
   }
}

// ... additional properties

]
"ClientContext":
{
    "<key>":<value">,
    \\ additional key value pairs
}
}
```

3.0 Deploy the EDS to WebSphere

Login to the WebSphere console: http://localhost:9060/ibm/console
Username: p8admin
password: filenet


localhost
https://localhost:9043/ibm/console/logon.jsp
Most Visited
Getting Started
Latest Headlines
Integrated Solutions Console
Log in

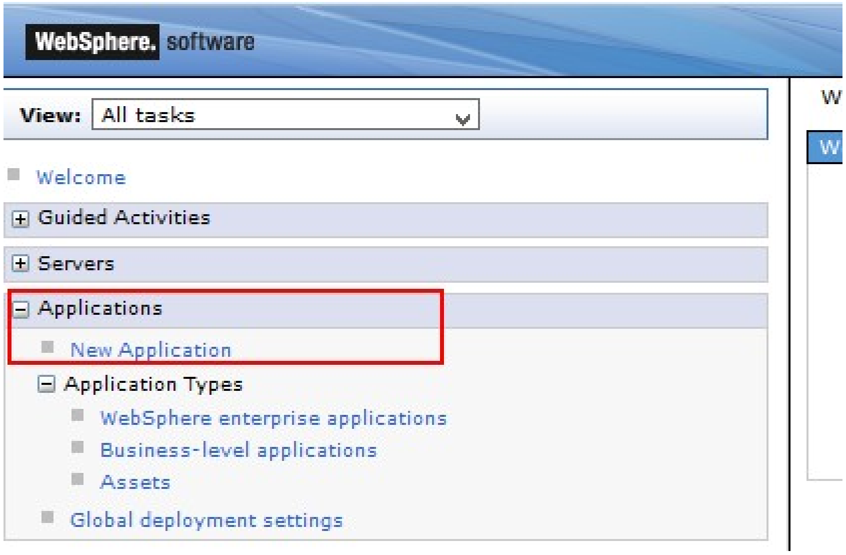
Integrated Solutions Console

Log in to the console.

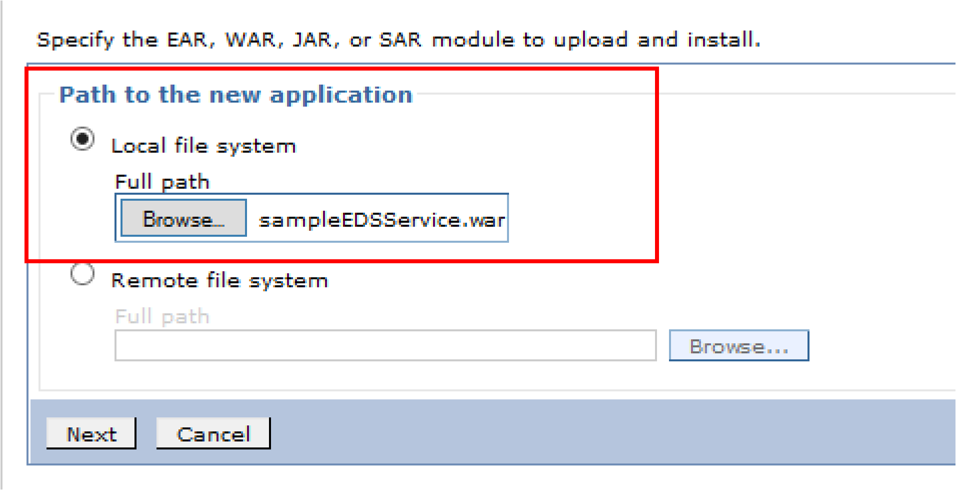
User ID: p8admin

Password: •••••••

Install a new WebSphere enterprise application:



Browse to the C:\External Data Integration folder and select the sampleEDSService.war file and click Next



Specify the EAR, WAR, JAR, or SAR module to upload and install.

Path to the new application

● Local file system

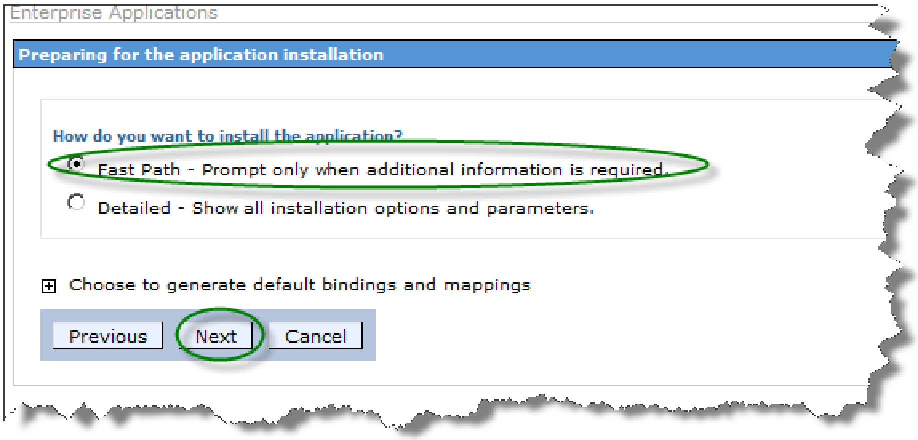
Full path

sampleEDSService.war

○ Remote file system

Full path

Keep the Fast Path default and click Next:



Enterprise Applications

Preparing for the application installation

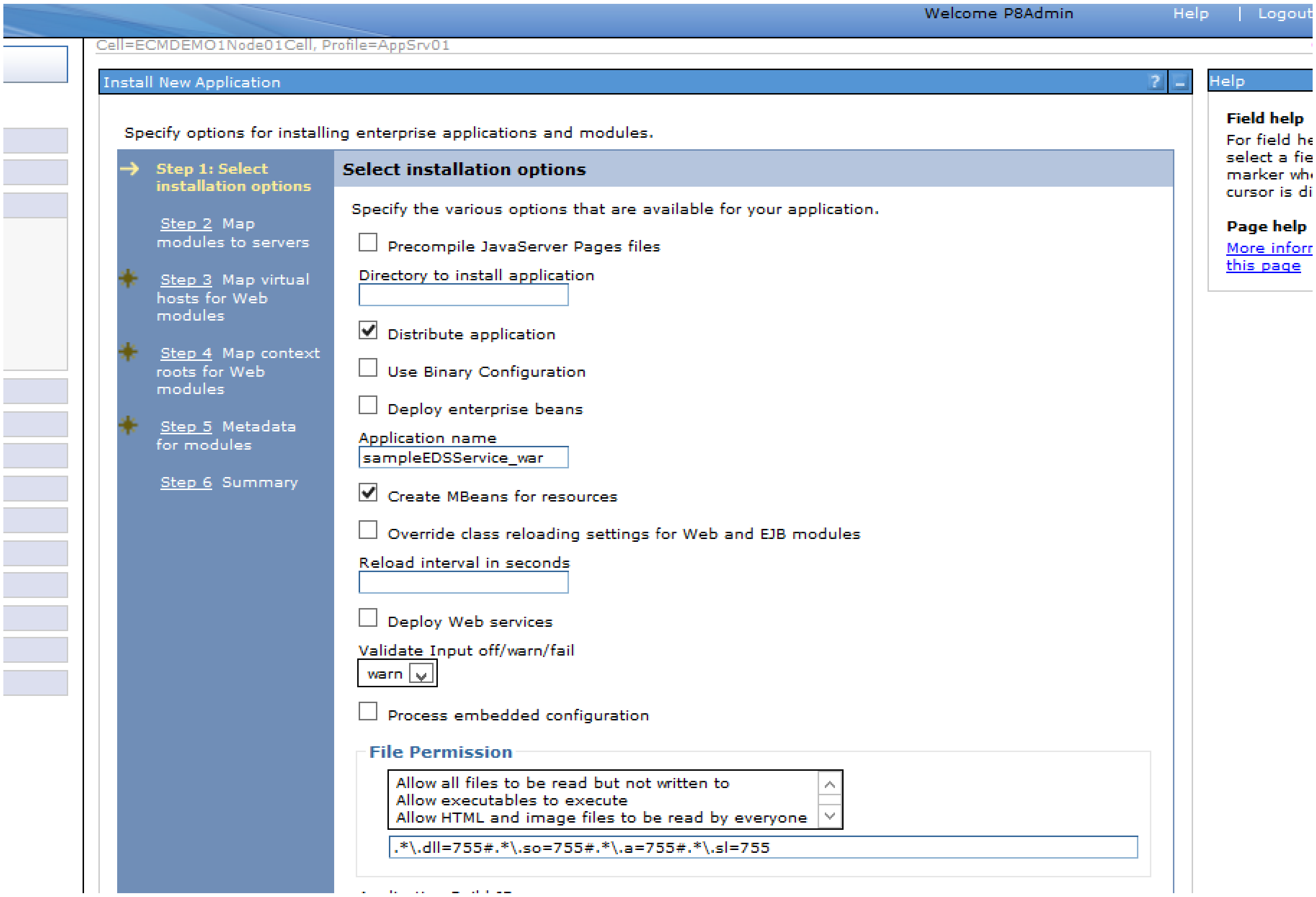
How do you want to install the application?

● Fast Path - Prompt only when additional information is required.

○ Detailed - Show all installation options and parameters.

⊞ Choose to generate default bindings and mappings

Keep defaults on **Step 1: Select installation options** and click Next:



Welcome P8Admin
Help | Logout

Cell=ECMDEMO1Node01Cell, Profile=AppSrv01

Install New Application

Specify options for installing enterprise applications and modules.

Step 1: Select installation options
Step 2 Map modules to servers
Step 3 Map virtual hosts for Web modules
Step 4 Map context roots for Web modules
Step 5 Metadata for modules
Step 6 Summary

Select installation options

Specify the various options that are available for your application.

□ Precompile JavaServer Pages files

Directory to install application

☑ Distribute application

□ Use Binary Configuration

□ Deploy enterprise beans

Application name: sampleEDSService_war

☑ Create MBeans for resources

□ Override class reloading settings for Web and EJB modules

Reload interval in seconds

□ Deploy Web services

Validate Input off/warn/fail: warn

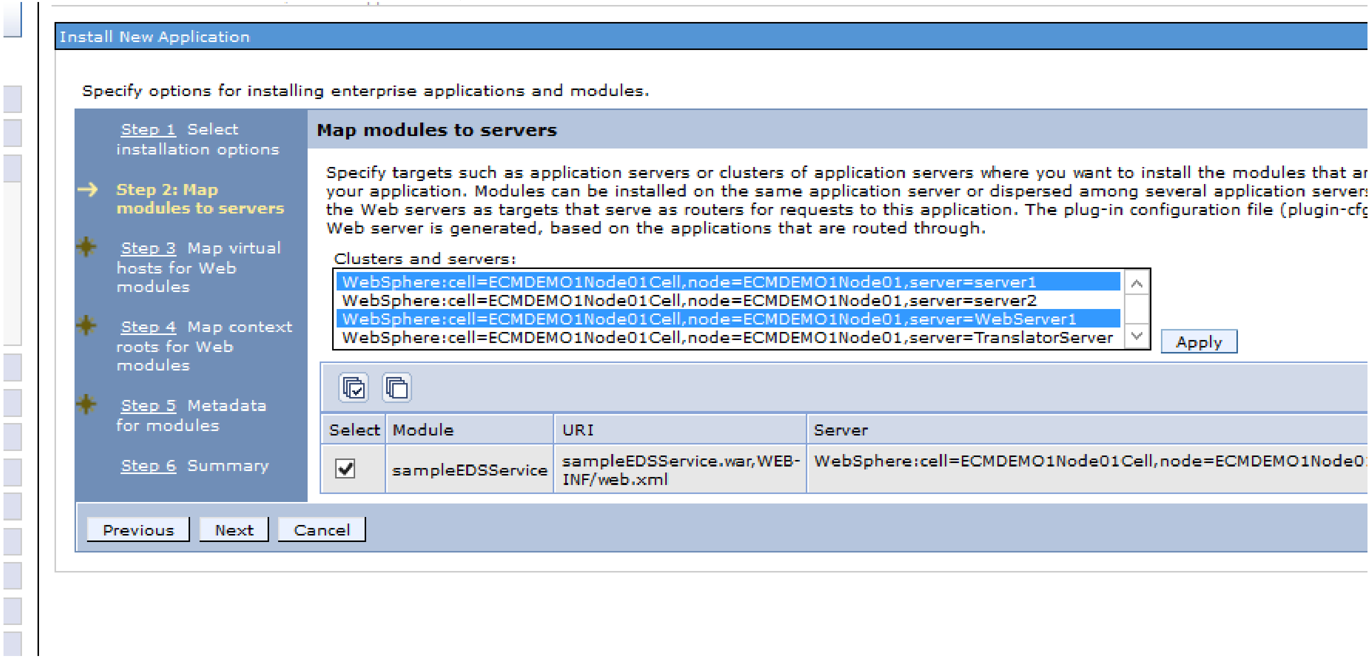
□ Process embedded configuration

File Permission

Allow all files to be read but not written to
Allow executables to execute
Allow HTML and image files to be read by everyone

.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755

Select server1 and WebServer1, check the module and click Next:



Install New Application

Specify options for installing enterprise applications and modules.

Step 1 Select installation options
→ Step 2: Map modules to servers
Step 3 Map virtual hosts for Web modules
Step 4 Map context roots for Web modules
Step 5 Metadata for modules
Step 6 Summary

Map modules to servers

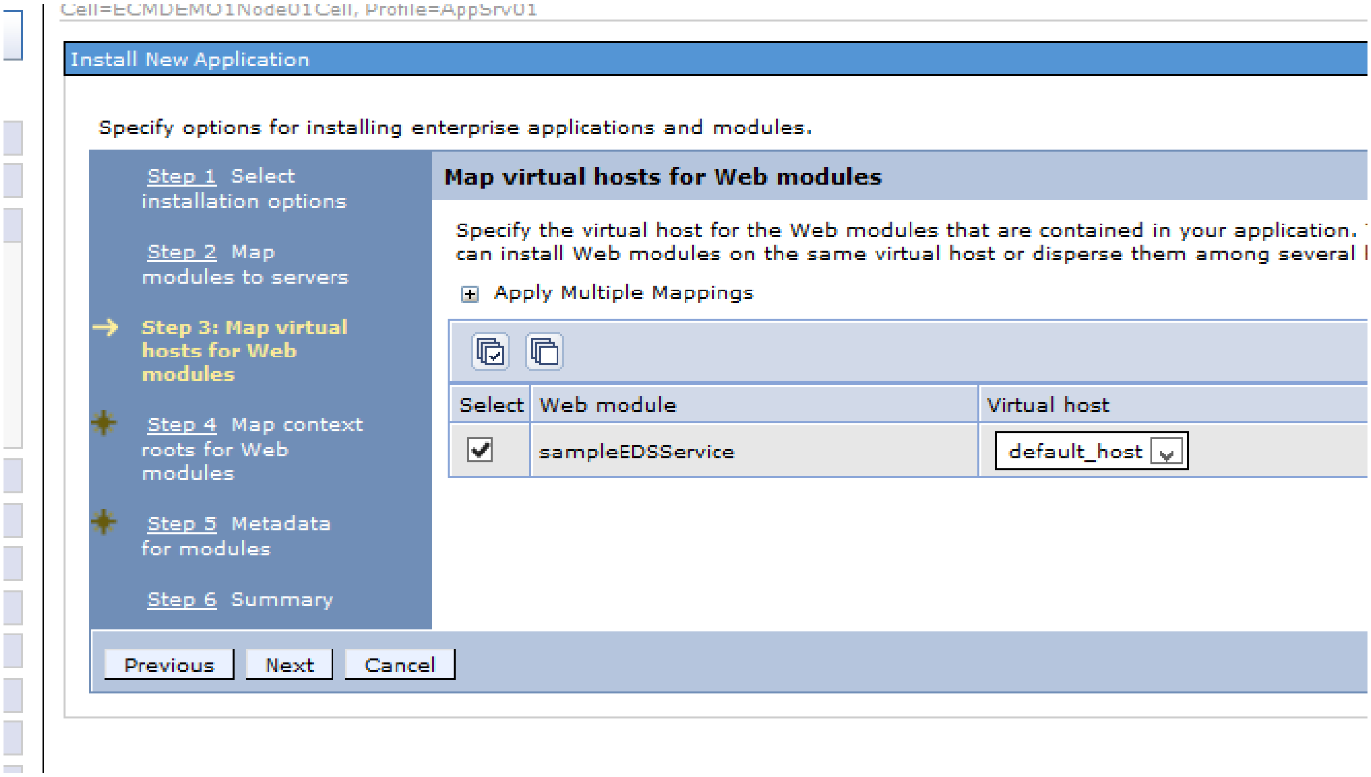
Specify targets such as application servers or clusters of application servers where you want to install the modules that ar
your application. Modules can be installed on the same application server or dispersed among several application server:
the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg
Web server is generated, based on the applications that are routed through.

Clusters and servers:

```
WebSphere:cell=ECMDEMO1Node01Cell,node=ECMDEMO1Node01,server=server1
WebSphere:cell=ECMDEMO1Node01Cell,node=ECMDEMO1Node01,server=server2
WebSphere:cell=ECMDEMO1Node01Cell,node=ECMDEMO1Node01,server=WebServer1
WebSphere:cell=ECMDEMO1Node01Cell,node=ECMDEMO1Node01,server=TranslatorServer
```

Select	Module	URI	Server
☑	sampleEDSService	sampleEDSService.war,WEB-INF/web.xml	WebSphere:cell=ECMDEMO1Node01Cell,node=ECMDEMO1Node0

Select the web module and click Next:



Cell=ECMDEMO1Node01Cell, Profile=AppSrv01

Install New Application

Specify options for installing enterprise applications and modules.

Step 1 Select installation options
Step 2 Map modules to servers
Step 3: Map virtual hosts for Web modules
Step 4 Map context roots for Web modules
Step 5 Metadata for modules
Step 6 Summary

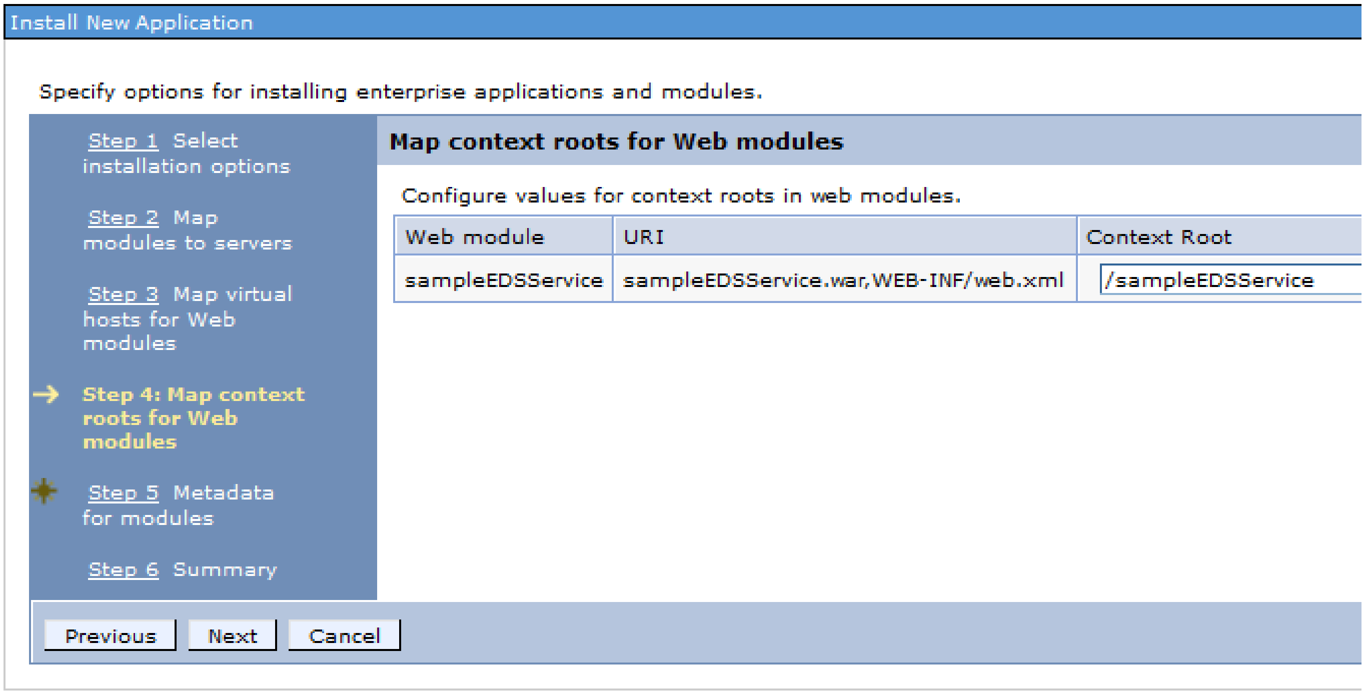
Map virtual hosts for Web modules

Specify the virtual host for the Web modules that are contained in your application. can install Web modules on the same virtual host or disperse them among several

⊞ Apply Multiple Mappings

Select	Web module	Virtual host
☑	sampleEDSService	default_host

Set the Context root to /sampleEDSService and click Next:



Install New Application

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

Step 2 Map modules to servers

Step 3 Map virtual hosts for Web modules

→ **Step 4: Map context roots for Web modules**

Step 5 Metadata for modules

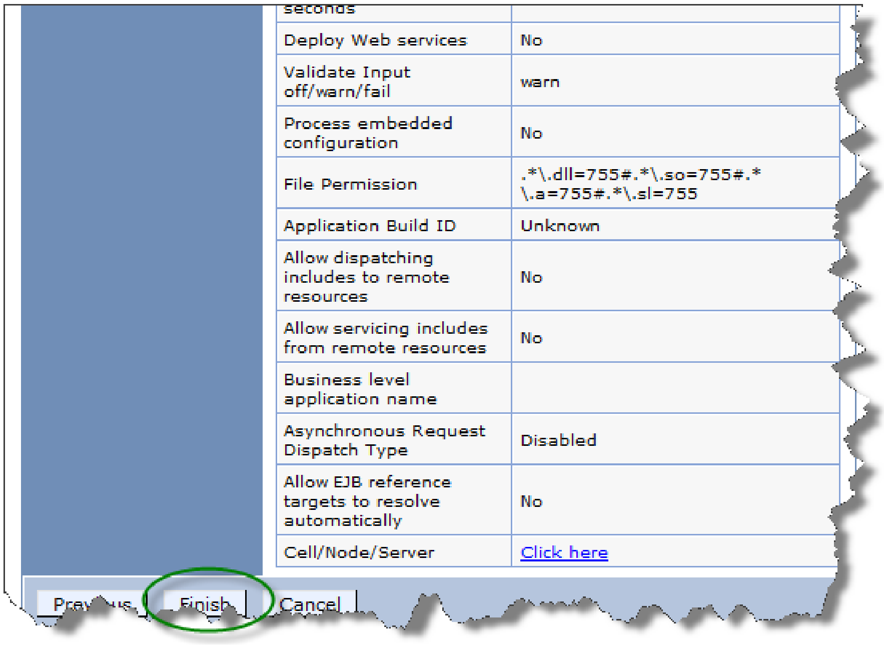
Step 6 Summary

Map context roots for Web modules

Configure values for context roots in web modules.

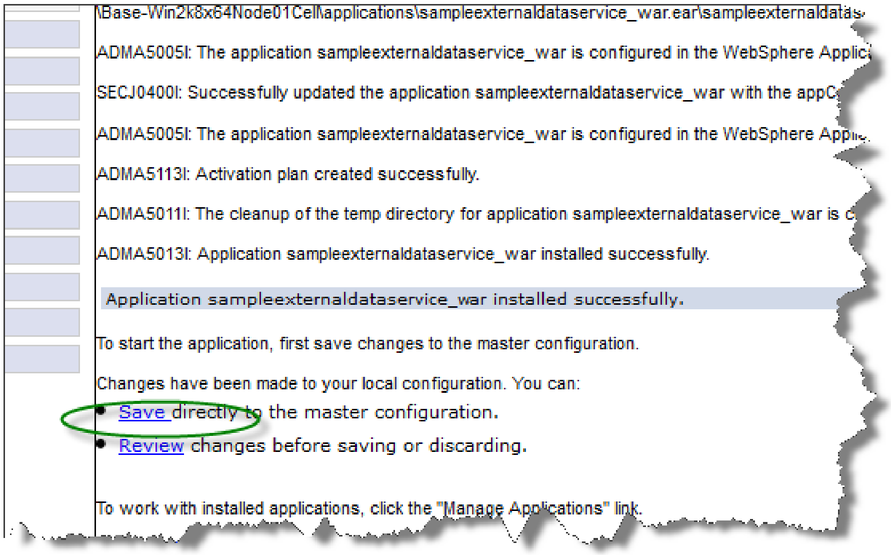
Web module	URI	Context Root
sampleEDSService	sampleEDSService.war,WEB-INF/web.xml	/sampleEDSService

Click Next (on the page that follows) to choose the defaults and then Click Finish on the Summary Page:



seconds	
Deploy Web services	No
Validate Input off/warn/fail	warn
Process embedded configuration	No
File Permission	.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755
Application Build ID	Unknown
Allow dispatching includes to remote resources	No
Allow servicing includes from remote resources	No
Business level application name	
Asynchronous Request Dispatch Type	Disabled
Allow EJB reference targets to resolve automatically	No
Cell/Node/Server	Click here

After successful deployment, Save your changes to the Master Configuration:



```
\Base-Win2k8x64Node01Cell\applications\sampleexternaldataservice_war.ear\sampleexternaldatas
ADMA5005I: The application sampleexternaldataservice_war is configured in the WebSphere Applic
SECJ0400I: Successfully updated the application sampleexternaldataservice_war with the appC
ADMA5005I: The application sampleexternaldataservice_war is configured in the WebSphere Appl
ADMA5113I: Activation plan created successfully.
ADMA5011I: The cleanup of the temp directory for application sampleexternaldataservice_war is c
ADMA5013I: Application sampleexternaldataservice_war installed successfully.
```

Application sampleexternaldataservice_war installed successfully.

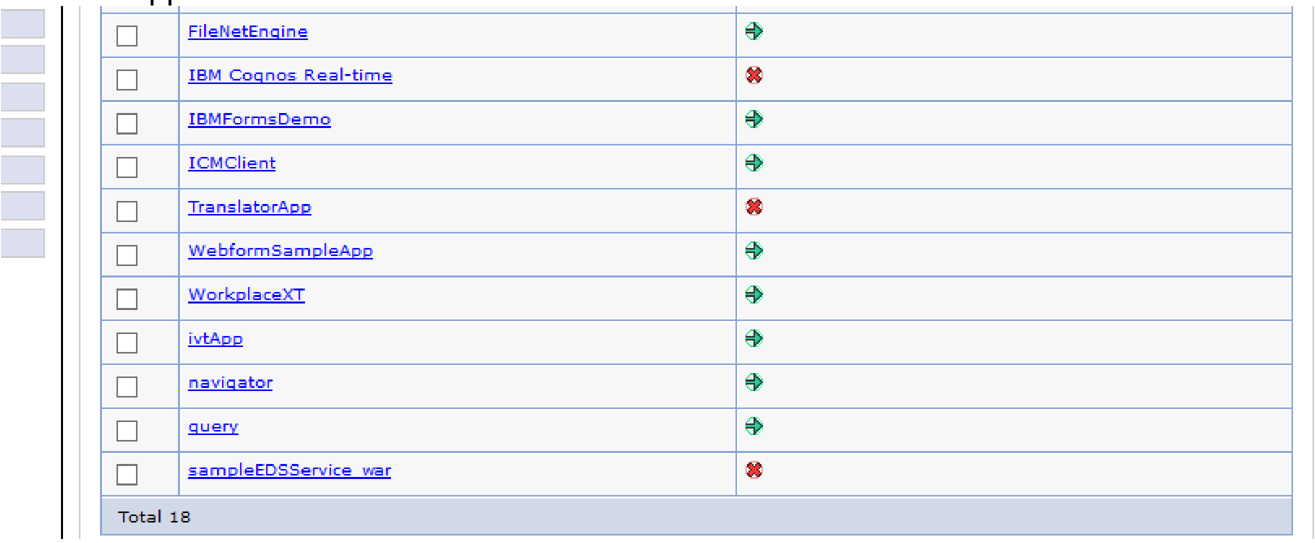
To start the application, first save changes to the master configuration.

Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

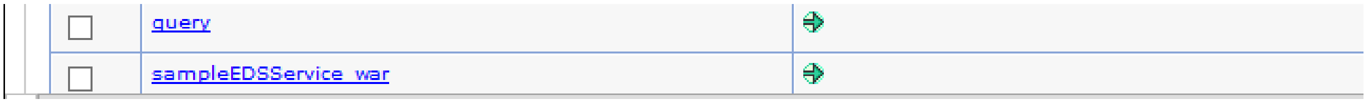
To work with installed applications, click the "Manage Applications" link.

Start the application:



□	FileNetEngine	→
□	IBM Cognos Real-time	✖
□	IBMFormsDemo	→
□	ICMClient	→
□	TranslatorApp	✖
□	WebformSampleApp	→
□	WorkplaceXT	→
□	ivtApp	→
□	navigator	→
□	query	→
□	sampleEDSService_war	✖
Total 18		

The application should now be started:

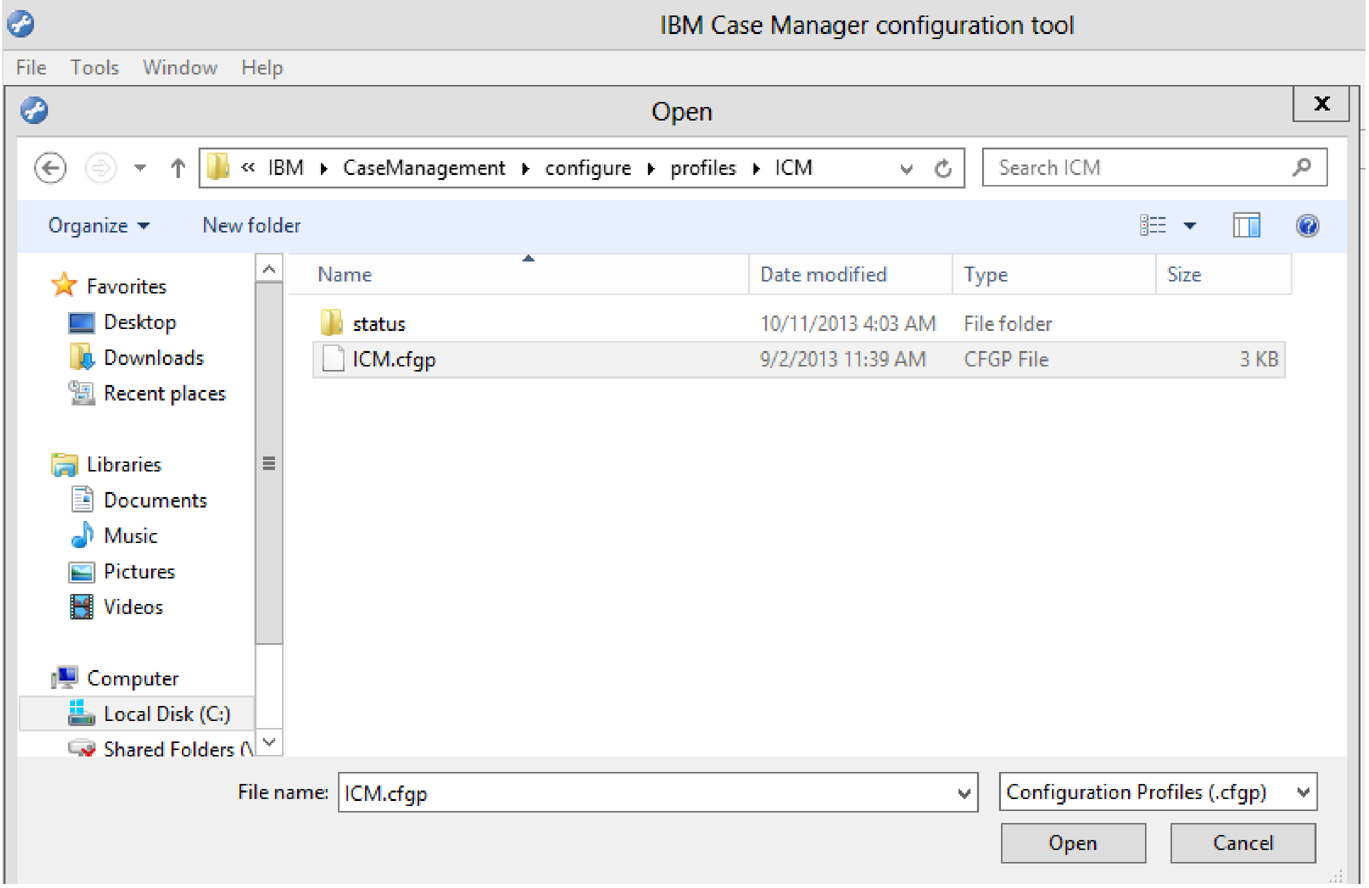


□	query	➔
□	sampleEDSService_war	➔

4.0 Register the EDS using CMAC

Launch Case Manager Configuration Tool from Start → All Programs → IBM Case Manager → Case Manager Configuration Tool

From File → Open Profile, open the profile ICM.cfgp in C:\IBM\CaseManagement\configure\profiles\ICM



« IBM ▸ CaseManagement ▸ configure ▸ profiles ▸ ICM

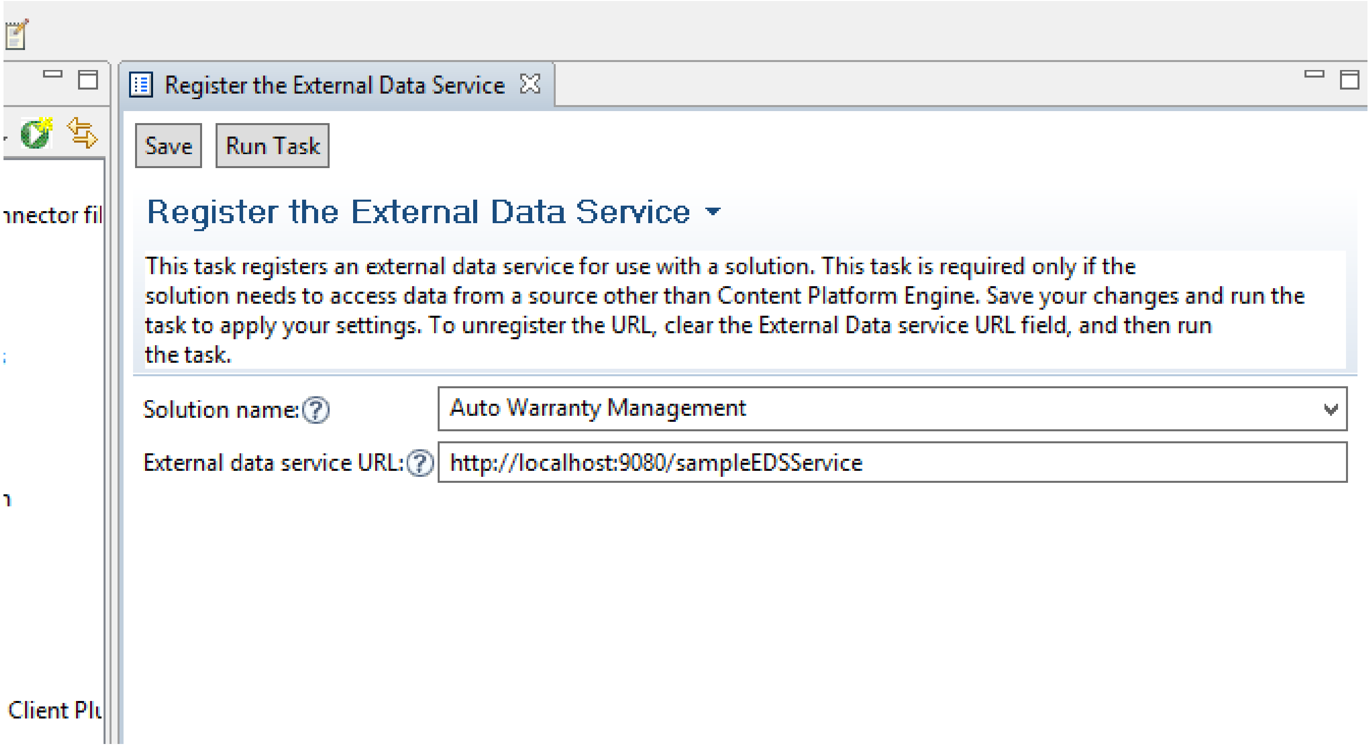
Name	Date modified	Type	Size
status	10/11/2013 4:03 AM	File folder	
ICM.cfgp	9/2/2013 11:39 AM	CFGP File	3 KB

File name: ICM.cfgp

Configuration Profiles (.cfgp)

Expand the ICM profile and open the “Register External Data Service” task by right clicking on it and selecting Edit Selected Task:

Select the Auto Warranty Management solution and enter the following URL in the External data service URL field:
http://localhost:9080/sampleEDSService



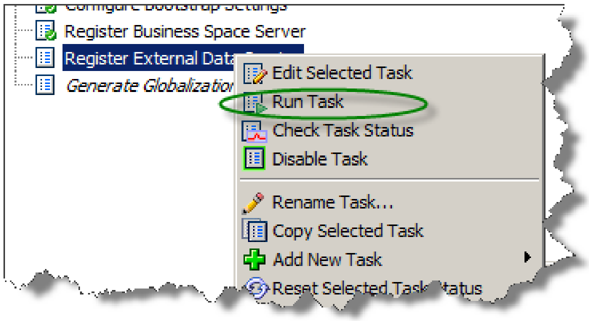
Register the External Data Service

This task registers an external data service for use with a solution. This task is required only if the solution needs to access data from a source other than Content Platform Engine. Save your changes and run the task to apply your settings. To unregister the URL, clear the External Data service URL field, and then run the task.

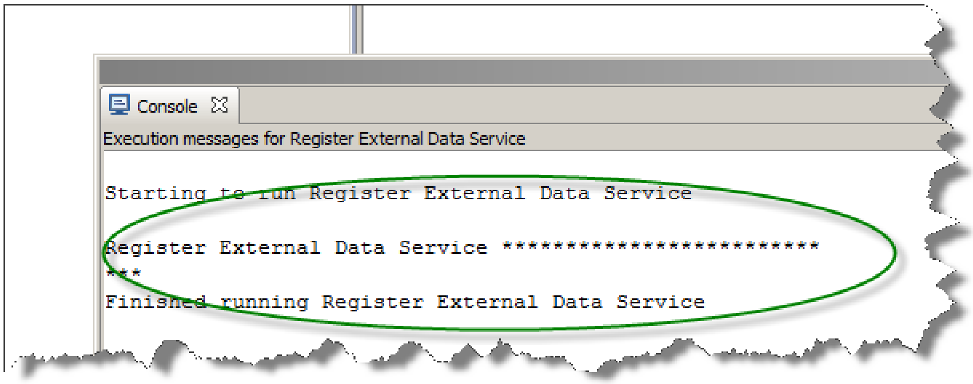
Solution name: Auto Warranty Management

External data service URL: http://localhost:9080/sampleEDSService

Save the task and then Run it:



After successful registration of the EDS, you can use it in the case management solution:



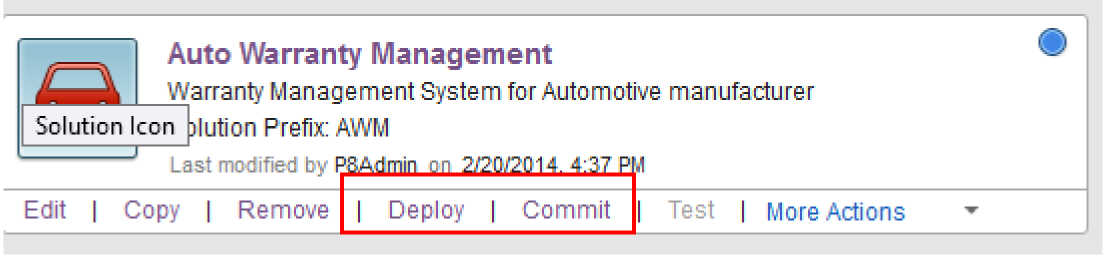
Execution messages for Register External Data Service

```
Starting to run Register External Data Service

Register External Data Service *************************
***
Finished running Register External Data Service
```

Exit Case Manager Configuration Tool and Restart the WebSphere Application Server (This is required only the first time the EDS application is deployed. Any further changes will require restarting the EDS application and the Navigator application in the WebSphere console)

Commit and Deploy the Auto Warranty Management solution from the Case Builder:



Auto Warranty Management

Warranty Management System for Automotive manufacturer

Solution Prefix: AWM

Last modified by P8Admin on 2/20/2014, 4:37 PM

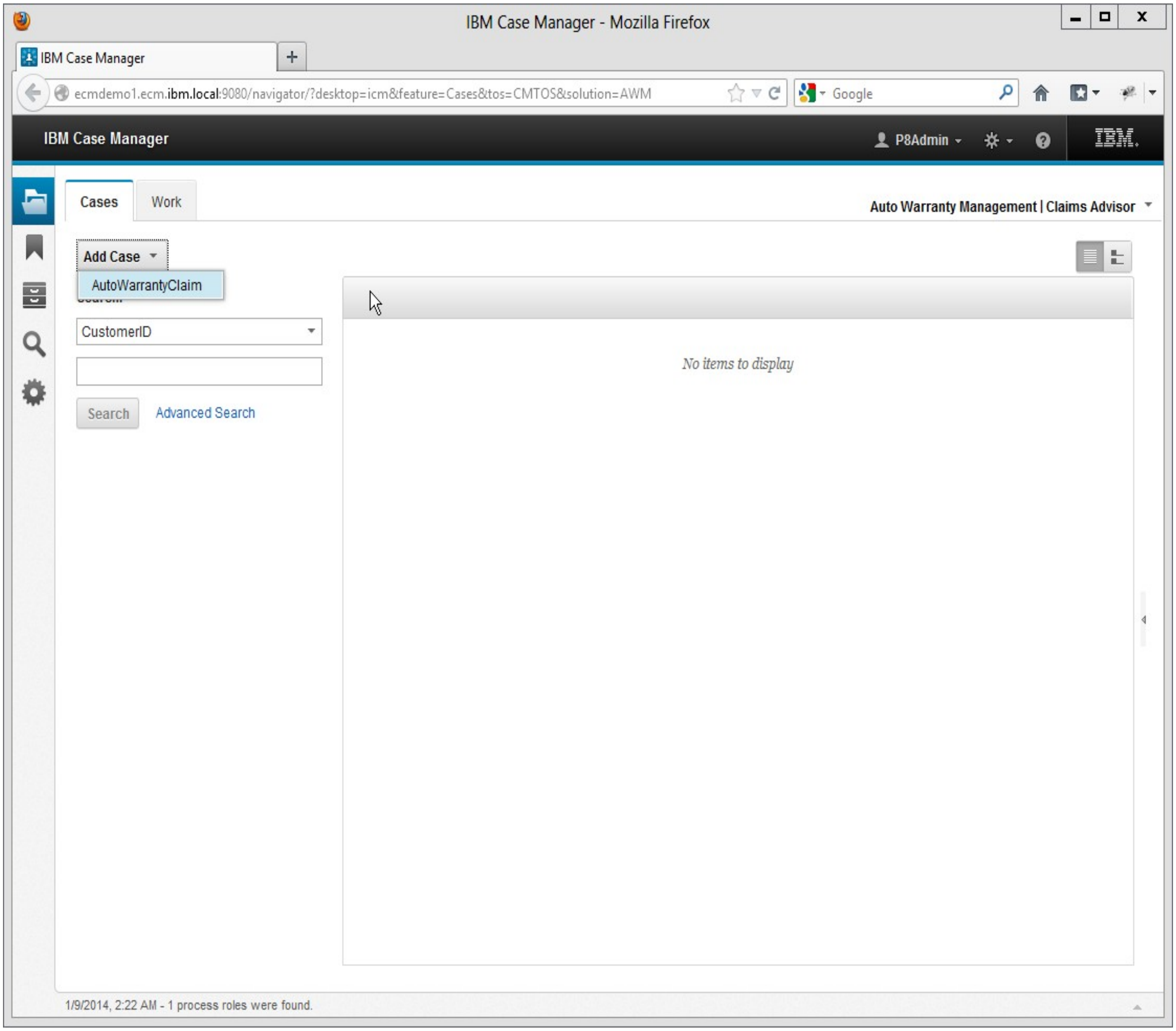
Solution Icon
Edit | Copy | Remove | Deploy | Commit | Test | More Actions

5.0 Testing the External Data Service

5.1 Create a new case

Click on the Test link from the Case Builder for the Auto Warranty Management solution to launch the Case Client

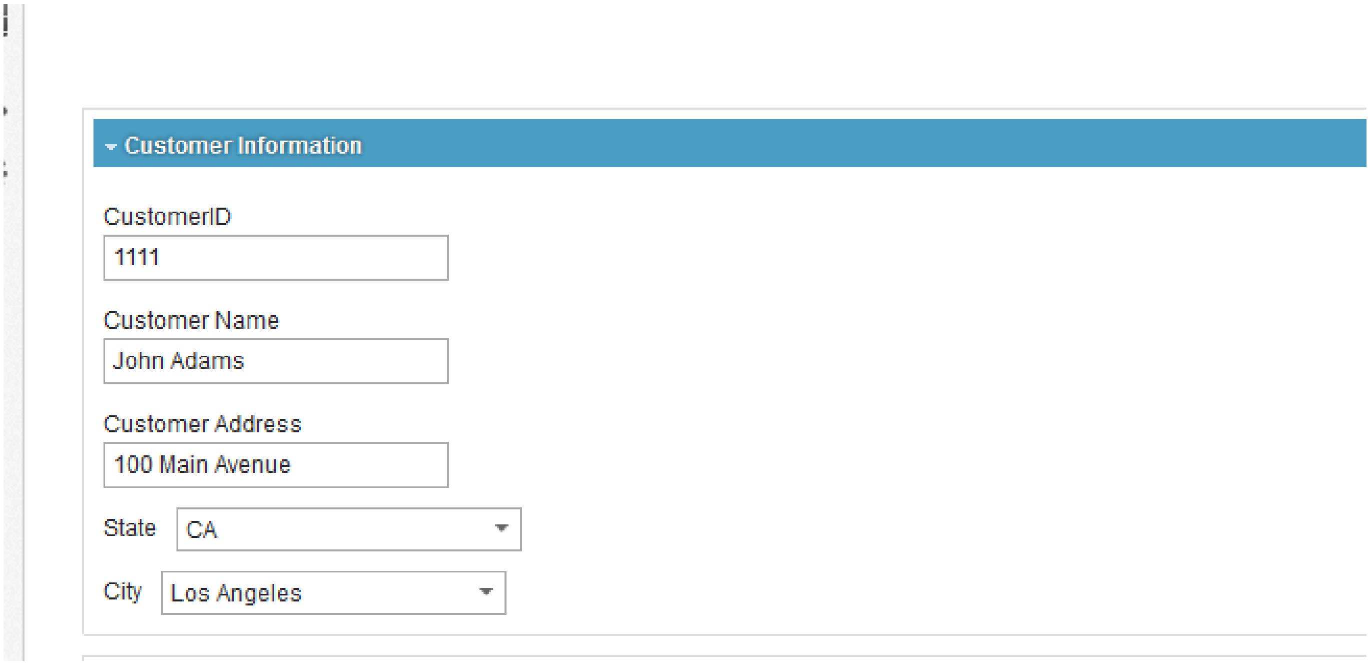
Add a new case instance with a case type of AutoWarrantyClaim:


IBM Case Manager - Mozilla Firefox
IBM Case Manager
ecmdemo1.ecm.ibm.local:9080/navigator/?desktop=icm&feature=Cases&tos=CMTOS&solution=AWM
Google
IBM Case Manager
P8Admin
IBM.
Cases
Work
Auto Warranty Management | Claims Advisor
Add Case
AutoWarrantyClaim
CustomerID
Search
Advanced Search
No items to display
1/9/2014, 2:22 AM - 1 process roles were found.

Expand the Customer Information section and fill in the CustomerID value as 1111. Tab out of the CustomerID text box, and notice that the Customer Name and Customer Address fields are filled in as follows:

The Case Data widget publishes the update of the CustomerID field to the CASE REST API, which in turn makes the call to the EDS to retrieve the dependent case properties such as Customer Name, Customer Address and the State choice list. Select a state from the State Choice list and notice that the widget publishes another event, which makes a call to the EDS through the CASE REST API and retrieves the dependent City choice list:

Expand the Vehicle section of Case Data and enter the value JA9999 in the VIN field. Tab out of the VIN field and the vehicle details will be filled out from values retrieved from the EDS. Notice that the EDS can designate case fields as required or optional, read-only or read-write, etc. For a list of properties returned by the EDS, review the JSON payload.



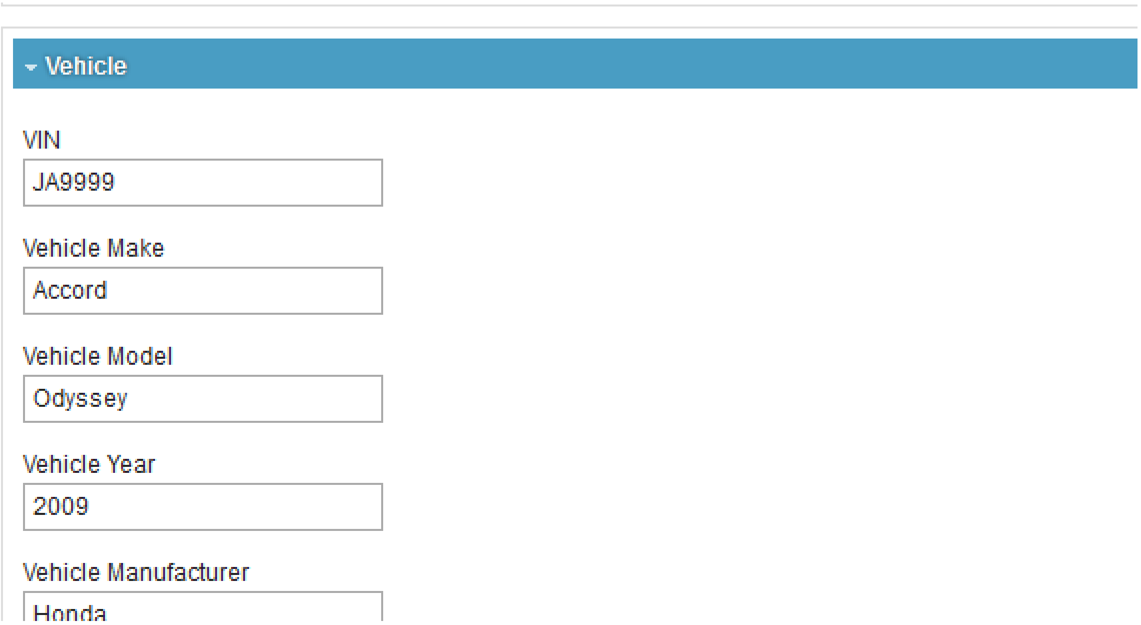
CustomerID
1111

Customer Name
John Adams

Customer Address
100 Main Avenue

State CA

City Los Angeles



VIN
JA9999

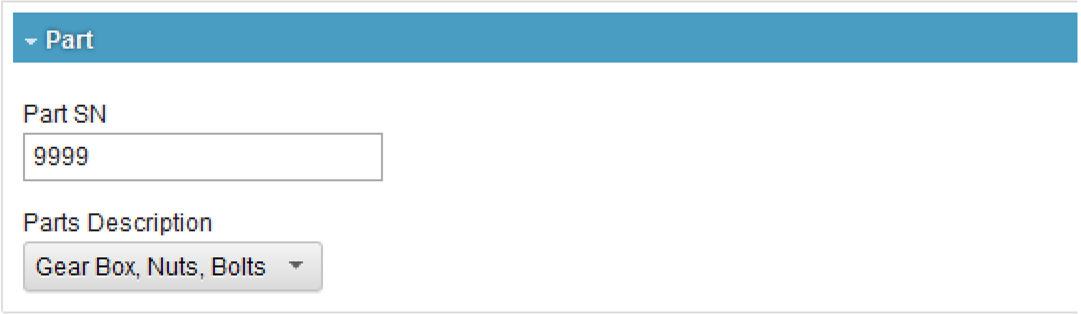
Vehicle Make
Accord

Vehicle Model
Odyssey

Vehicle Year
2009

Vehicle Manufacturer
Honda

Expand the Part section of Case Data and enter the value 9999 in the Part SN field, and hit the enter key. The Parts Description is a Multi-Value string field. The EDS returns a list of values for the Parts Description. Select the parts description, clicking on the Parts Description link as show below:

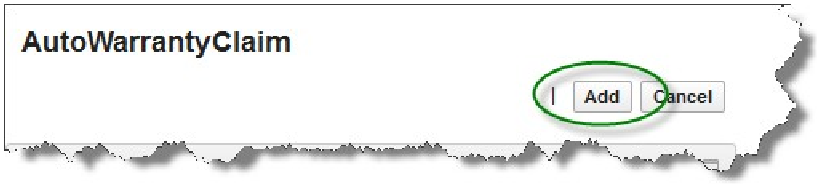


Part

Part SN: 9999

Parts Description: Gear Box, Nuts, Bolts

Click OK and Add the case



Now, let's take a look at the REST POST HTTP requests made by the Case Manager REST API to the EDS and the corresponding JSON responses from the EDS merged by the Case Manager. Open the SystemOut.log file from C:\IBM\WebSphere\AppServer\profiles\AppSrv01\logs\server1

The first request sent to the EDS is with requestMode=initialNewObject as shown in the JSONpayload below.

This is the request payload sent to the EDS from Case Manager. It includes all the case properties (including system properties), that the EDS can populate in the response.

```
sampleEDService.UpdateObjectTypeServlet: objectType=AWM_AutoWarrantyClaim
requestMode=initialNewObject userid=null desktop=null locale=null

[1/9/14 2:55:37:205 PST] 000000e7 SystemOut     O
{"requestMode":"initialNewObject","properties":[{"value":null,"symbolicName":"Creator"},
{"value":null,"symbolicName":"DateCreated"},{"value":null,"symbolicName":"LastModifier"},
{"value":null,"symbolicName":"DateLastModified"},{"value":null,"symbolicName":"Id"},
{"value":null,"symbolicName":"Name"},{"value":null,"symbolicName":"Owner"},
{"value":null,"symbolicName":"LockToken"},{"value":null,"symbolicName":"LockTimeout"},
{"value":null,"symbolicName":"LockOwner"},{"value":null,"symbolicName":"FolderName"},
{"value":null,"symbolicName":"PathName"},
{"value":true,"symbolicName":"InheritParentPermissions"},
{"value":null,"symbolicName":"IndexationId"},
{"value":null,"symbolicName":"CmIndexingFailureCode"},
{"value":null,"symbolicName":"CmRetentionDate"},
{"value":false,"symbolicName":"CmIsMarkedForDeletion"},
{"value":false,"symbolicName":"IsHiddenContainer"},
{"value":null,"symbolicName":"ContainerType"},
{"value":null,"symbolicName":"ClbCreatorIndex"},
{"value":null,"symbolicName":"ClbLastModifierIndex"},
{"value":null,"symbolicName":"CmAcmCaseIdentifier"},
{"value":0,"symbolicName":"CmAcmCaseState"},
{"value":"http:\/\/localhost:9080\/sampleEDSService","symbolicName":"CmAcmExternalDataURI
"},{"value":[],"symbolicName":"CmAcmDocumentFilingInstruction"},
{"value":null,"symbolicName":"AWM_CustomerID"},
{"value":null,"symbolicName":"AWM_VehicleYear"},{"value":
[],"symbolicName":"AWM_PartsDescription"},
{"value":null,"symbolicName":"AWM_VehicleModel"},
{"value":null,"symbolicName":"AWM_State"},
```

{"value":null,"symbolicName":"AWM_ProductionPlant"},
{"value":null,"symbolicName":"AWM_CustomerAddress"},
{"value":null,"symbolicName":"AWM_CustomerName"},
{"value":null,"symbolicName":"AWM_VehicleMake"},{"value":null,"symbolicName":"AWM_VIN"},
{"value":null,"symbolicName":"AWM_City"},
{"value":null,"symbolicName":"AWM_VehicleManufacturer"},
{"value":null,"symbolicName":"AWM_PartSN"},
{"value":null,"symbolicName":"AWM_CountryManufacturer"}],"clientContext":
{"currentSolution":"Auto Warranty Management","role":"Claims
Advisor"},"repositoryId":"CMTOS"}{"properties":
[{"value":null,"hasDependentProperties":true,"symbolicName":"AWM_VIN"},
{"value":null,"hasDependentProperties":true,"symbolicName":"AWM_CustomerID"},
{"value":null,"hasDependentProperties":true,"symbolicName":"AWM_PartSN"}],"externalDataIdentifier":"2900998"}AWM_AutoWarrantyClaim

In the AWM_AutoWarrantyClaim_PropertyData.json file, we have set up the JSON values for AWM_CustomerID to be a dependant property, so when the user enters a value (1111), the corresponding dependant values such as CustomerName will be populated.

When the user enters the value for CustomerID, the Case Client sends another POST request to the EDS, with the following payload with requestMode=inProgressChanges. The response will include all the dependant values for CustomerName, CustomerAddress etc.

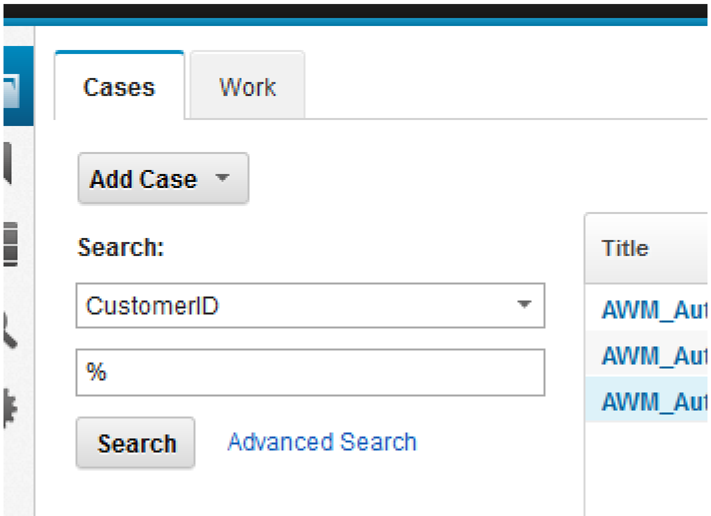
{"requestMode":"inProgressChanges","properties":[{"value":null,"symbolicName":"Creator"},
{"value":null,"symbolicName":"DateCreated"},{"value":null,"symbolicName":"LastModifier"},
{"value":null,"symbolicName":"DateLastModified"},{"value":null,"symbolicName":"Id"},
{"value":null,"symbolicName":"Name"},{"value":null,"symbolicName":"Owner"},
{"value":null,"symbolicName":"LockToken"},{"value":null,"symbolicName":"LockTimeout"},
{"value":null,"symbolicName":"LockOwner"},{"value":null,"symbolicName":"FolderName"},
{"value":null,"symbolicName":"PathName"},
{"value":true,"symbolicName":"InheritParentPermissions"},
{"value":null,"symbolicName":"IndexationId"},
{"value":null,"symbolicName":"CmIndexingFailureCode"},
{"value":null,"symbolicName":"CmRetentionDate"},
{"value":false,"symbolicName":"CmIsMarkedForDeletion"},
{"value":false,"symbolicName":"IsHiddenContainer"},
{"value":null,"symbolicName":"ContainerType"},
{"value":null,"symbolicName":"ClbCreatorIndex"},
{"value":null,"symbolicName":"ClbLastModifierIndex"},
{"value":null,"symbolicName":"CmAcmCaseIdentifier"},
{"value":0,"symbolicName":"CmAcmCaseState"},
{"value":"http:\/\/localhost:9080\/sampleEDSService","symbolicName":"CmAcmExternalDataURI"},{"value":[],"symbolicName":"CmAcmDocumentFilingInstruction"},
{"value":"1111","symbolicName":"AWM_CustomerID"},
{"value":null,"symbolicName":"AWM_VehicleYear"},{"value":
[],"symbolicName":"AWM_PartsDescription"},
{"value":null,"symbolicName":"AWM_VehicleModel"},

{"value":null,"symbolicName":"AWM_State"},
{"value":null,"symbolicName":"AWM_ProductionPlant"},
{"value":null,"symbolicName":"AWM_CustomerAddress"},
{"value":null,"symbolicName":"AWM_CustomerName"},
{"value":null,"symbolicName":"AWM_VehicleMake"},{"value":null,"symbolicName":"AWM_VIN"},
{"value":null,"symbolicName":"AWM_City"},
{"value":null,"symbolicName":"AWM_VehicleManufacturer"},
{"value":null,"symbolicName":"AWM_PartSN"},
{"value":null,"symbolicName":"AWM_CountryManufacturer"}],"clientContext":
{"currentSolution":"Auto Warranty Management","role":"Claims
Advisor"},"repositoryId":"CMTOS","externalDataIdentifier":"2900998"}{"properties":
[{"value":null,"hasDependentProperties":true,"symbolicName":"AWM_VIN"},
{"value":"1111","hasDependentProperties":true,"symbolicName":"AWM_CustomerID"},
{"value":"John
Adams","dependentValue":"1111","symbolicName":"AWM_CustomerName","dependentOn":"AW
M_CustomerID"},{"value":"100 Main
Avenue","dependentValue":"1111","symbolicName":"AWM_CustomerAddress","dependentOn":"
AWM_CustomerID"},
{"dependentValue":"1111","hasDependentProperties":true,"symbolicName":"AWM_State","depen
dentOn":"AWM_CustomerID","choiceList":{"choices":[{"value":"CA","displayName":"CA"},
{"value":"NY","displayName":"NY"},{"value":"NV","displayName":"NV"},
{"value":"MS","displayName":"MS"}],"displayName":"States"}},
{"value":null,"hasDependentProperties":true,"symbolicName":"AWM_PartSN"}],"externalDataIde
ntifier":"2900998"}AWM_AutoWarrantyClaim

5.2 Modify a case

Now let's suppose that the Vehicle Information database was recently updated, which impacted the Vehicle Model case data field for the VIN 9999. We will first update the sample external data source with the new value for the Vehicle Model case data field, and when the case is opened, the updated value will be displayed in the Vehicle Model field.

Before making your changes, search for the previously created case by doing a case search on the Cases page. You can use % as a wildcard for the CustomerID.

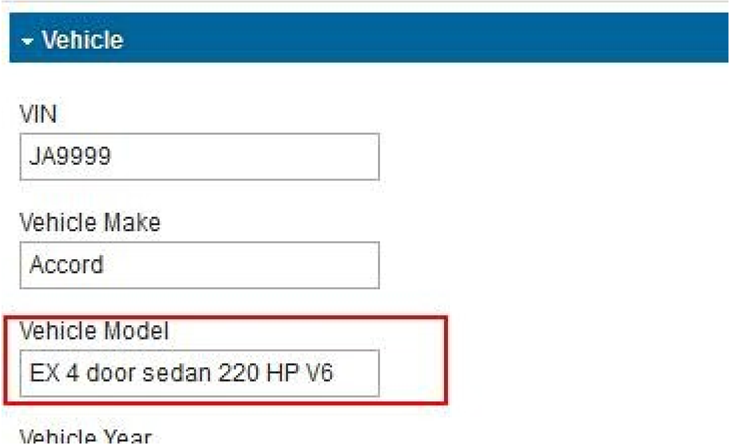


Search:

CustomerID

%

In the resulting Case List, click on the case link, and the Case Details page will be displayed. On that page the current Vehicle Model value EX 4 door sedan 220 HP V6 is shown in the Case Data widget:



VIN

JA9999

Vehicle Make

Accord

Vehicle Model

EX 4 door sedan 220 HP V6

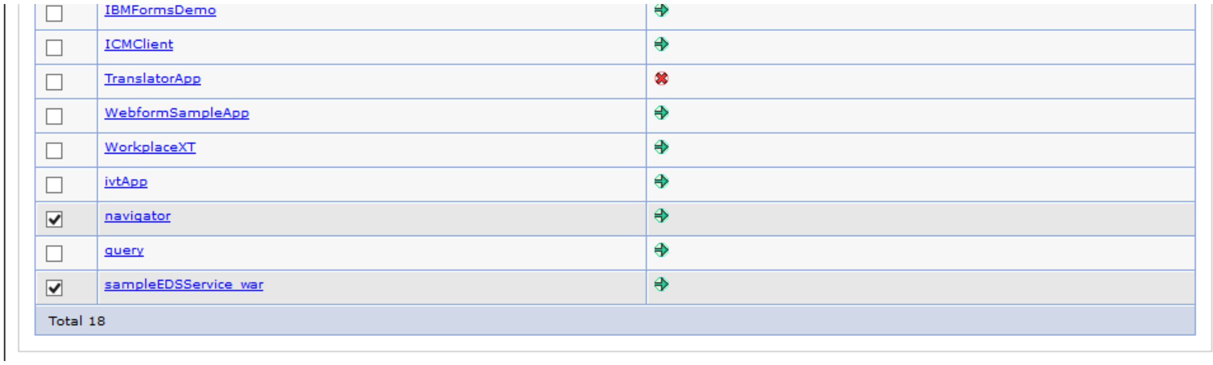
Vehicle Year

To modify the value in the EDS, browse to this folder:
C:\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\ECMDEMO1Node01Cell\sampleEDSService_war.ear\sampleEDSService.war\WEB-INF\classes

Rename the AWM_AutoWarrantyClaim_PropertyData.json file to AWM_AutoWarrantyClaim_PropertyData.orig

Copy the AWM_AutoWarrantyClaim_PropertyData.json from C:\External Data Integration folder to
C:\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\ECMDEMO1Node01Cell\sampleEDSService_war.ear\sampleEDSService.war\WEB-INF\classes

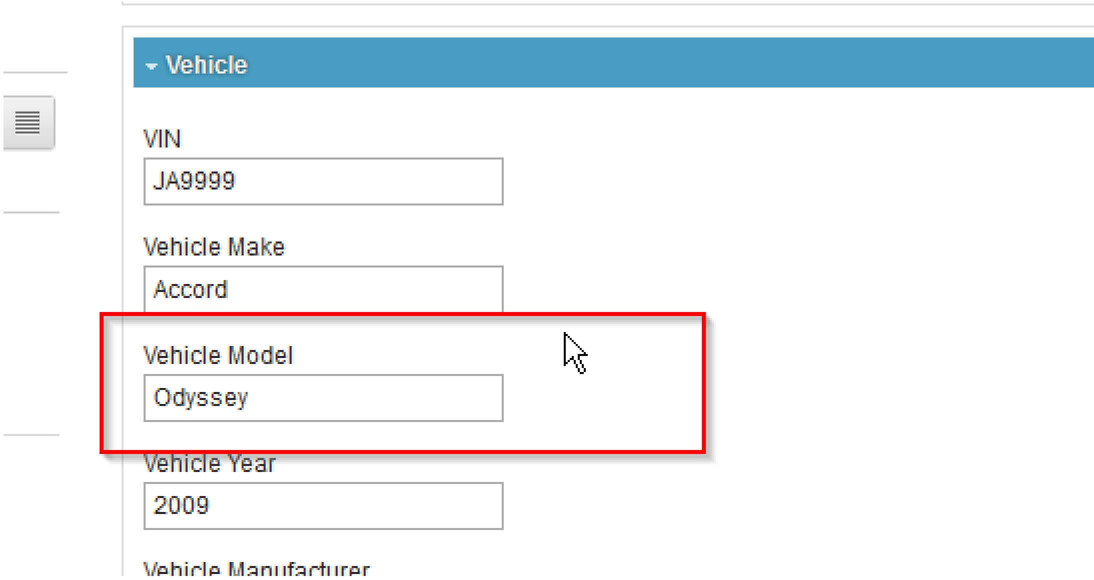
Login to the WebSphere console and stop and start the EDS and navigator applications as follows:
http://ecmdemo1:9060/ibm/console
p8admin/filenet



☐	IBMFormsDemo	➡
☐	ICMClient	➡
☐	TranslatorApp	✖
☐	WebformSampleApp	➡
☐	WorkplaceXT	➡
☐	ivtApp	➡
☑	navigator	➡
☐	query	➡
☑	sampleEDSService_war	➡
Total 18		

Select the application, stop and start it. Close the browser and launch the Case Client again (login to Case Builder and select Test link from the Auto Warranty Management solution)

Search for the case again, and open it in the Case Details page. The value in the Vehicle Model case data field has changed. If you don't see the change, log out of Case Client and log back in, then search again.



Vehicle

VIN

JA9999

Vehicle Make

Accord

Vehicle Model

Odyssey

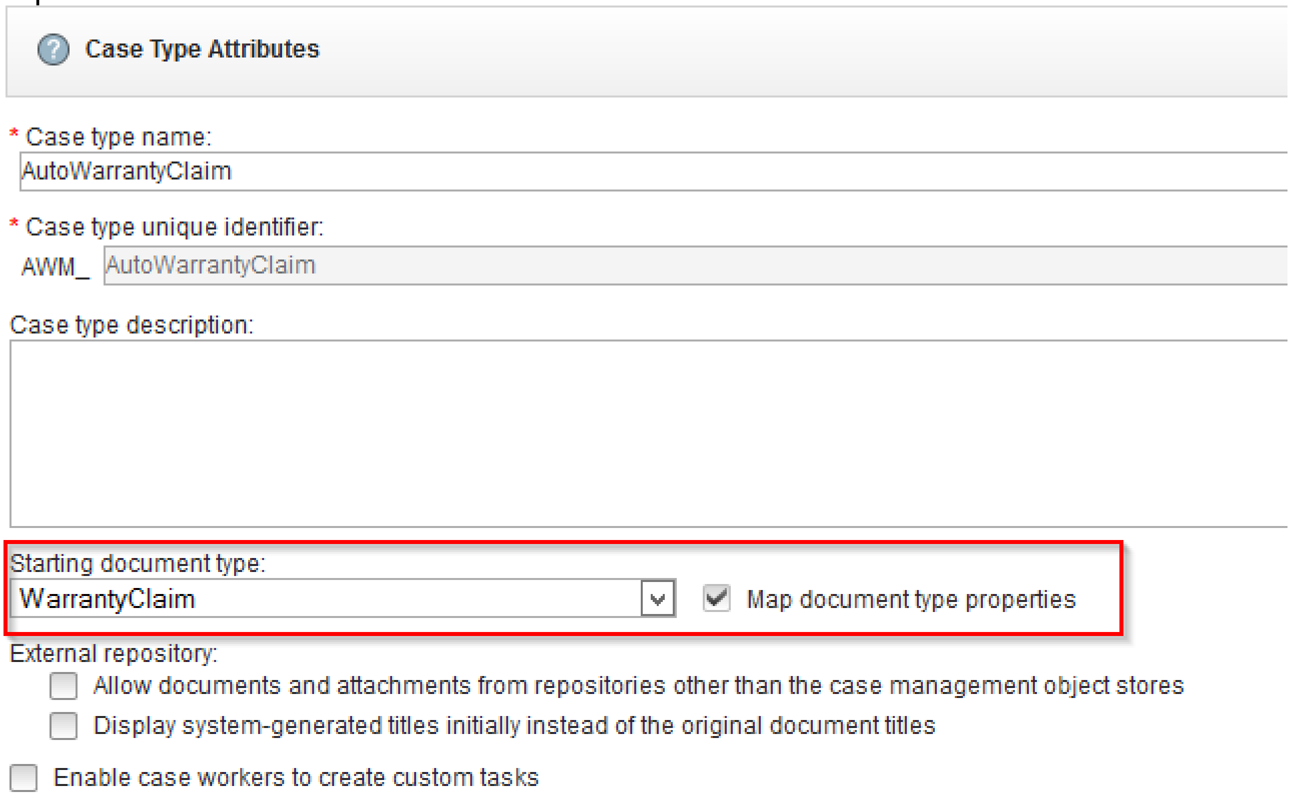
Vehicle Year

2009

Vehicle Manufacturer

5.3 Mapping Document Properties to Case Properties

The Auto Warranty Management solution has been configured to create a case whenever a document of type WarrantyClaim is created. You can map the document properties to the case properties as shown below:



Case Type Attributes

* Case type name:
AutoWarrantyClaim

* Case type unique identifier:
AWM_ AutoWarrantyClaim

Case type description:

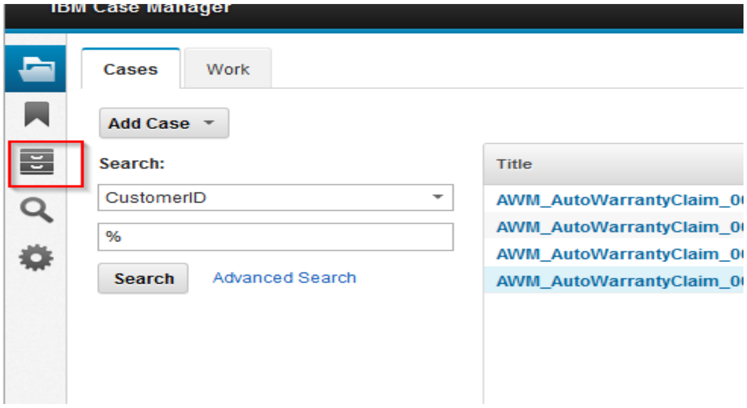
Starting document type:
WarrantyClaim ☑ Map document type properties

External repository:
□ Allow documents and attachments from repositories other than the case management object stores
□ Display system-generated titles initially instead of the original document titles

□ Enable case workers to create custom tasks

When you create a document of type WarrantyClaim and fill in the CustomerID, Part SN and VIN, a case will be created and the document will be moved into the case. When you open the case data, the external data will be populated using the CustomerID, Part SN and VIN.

Let's go through these steps now.
In the Case Client click on the Browse feature as shown below:



IBM Case Manager

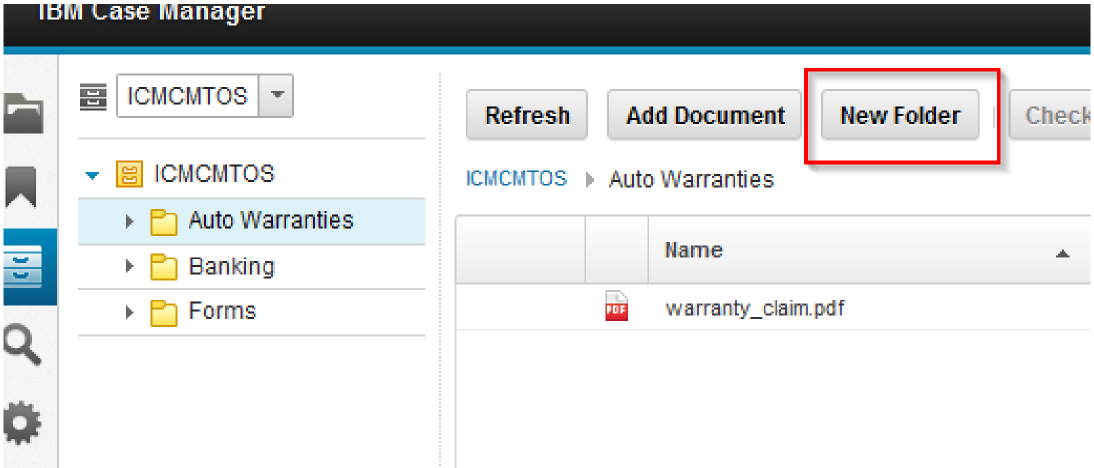
Cases Work

Add Case

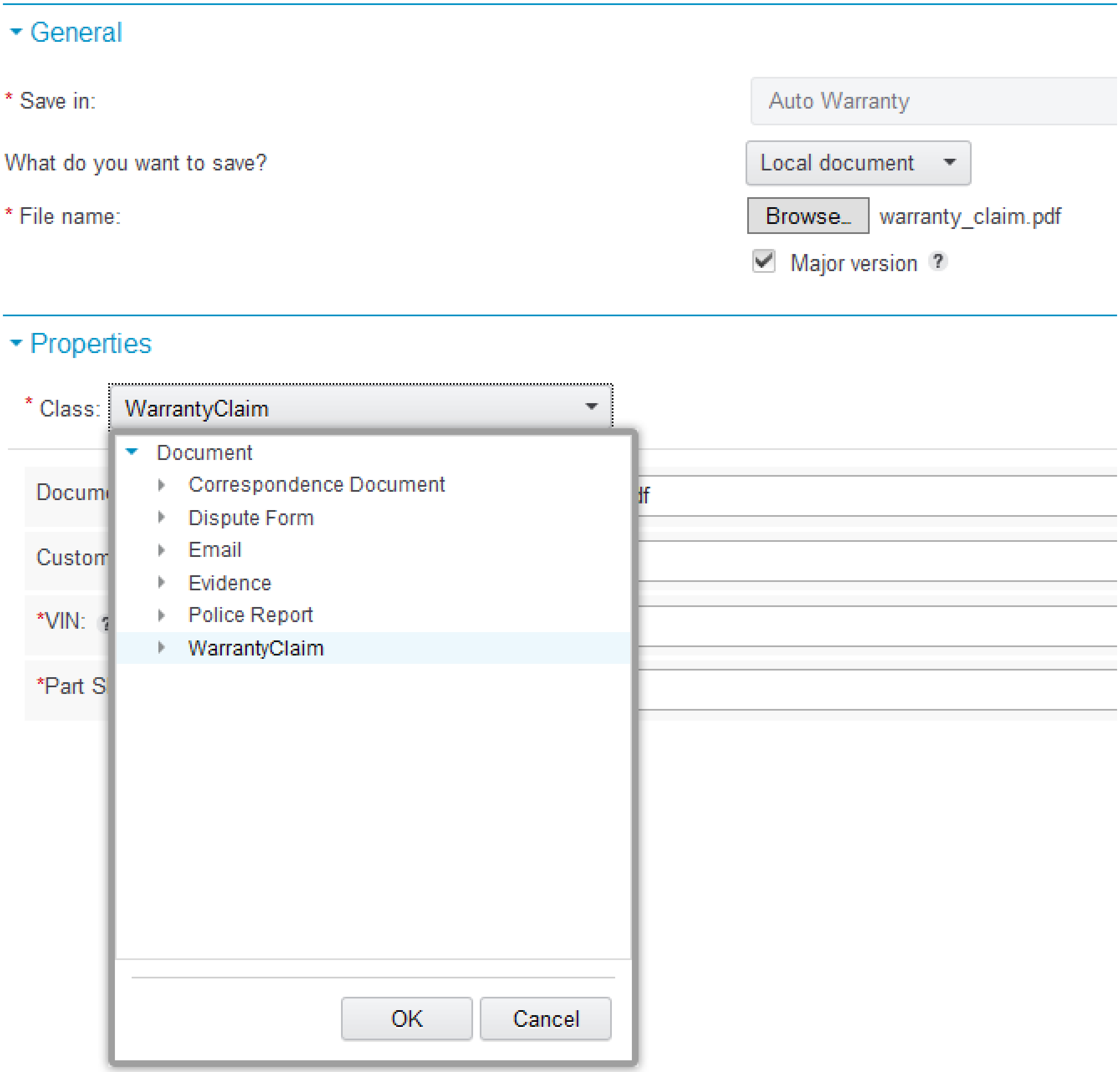
Search:
CustomerID
%
Search Advanced Search

Title
AWM_AutoWarrantyClaim_0
AWM_AutoWarrantyClaim_0
AWM_AutoWarrantyClaim_0
AWM_AutoWarrantyClaim_0

Create a folder named Auto Warranties



Browse to the new folder and drag and drop the warranty_claim.pdf document from C:\External Data Integration folder into the Auto Warranties folder in the Case Client. This will launch the Add Document dialog as shown below. Select the Warranty Claim as the document type:



General

* Save in: Auto Warranty

What do you want to save? Local document

* File name: Browse... warranty_claim.pdf

☑ Major version ?

Properties

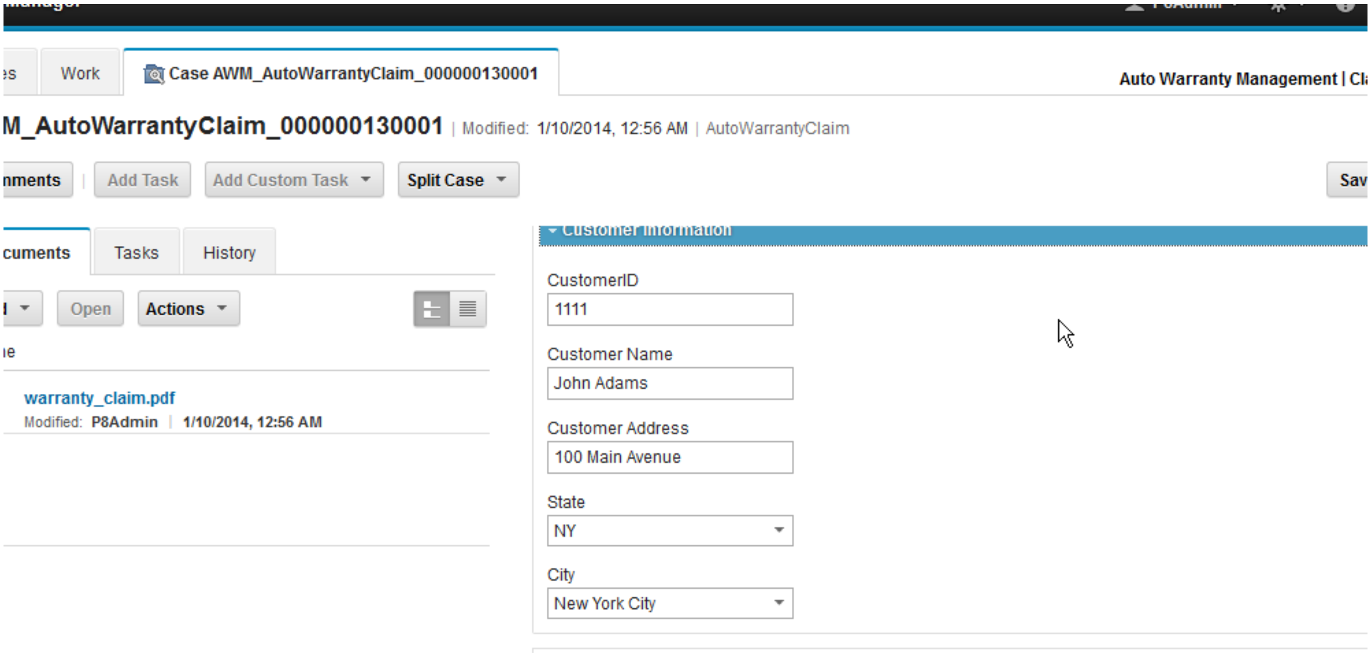
* Class: WarrantyClaim

Enter the document properties as follows:
Customer ID : 1111, tab out and you will see the Customer Name populated
VIN: JA9999, tab out and you will see the Vehicle make populated
Part SN: 9999, tab out and you will see the Parts Description populated

Click Add to create the document and therefore a new case.

Login to Case Client using p8admin/filenet credentials and search for the case on the Cases page. (Search based on the Date Added attribute and pick today's date).

Click on the case link to open the case in Case Details, and the case data will be retrieved from the external data service and populated. Select the State and City values in the Customer tab and save.



M_AutoWarrantyClaim_000000130001 | Modified: 1/10/2014, 12:56 AM | AutoWarrantyClaim

warranty_claim.pdf
Modified: P8Admin | 1/10/2014, 12:56 AM

CustomerID: 1111
Customer Name: John Adams
Customer Address: 100 Main Avenue
State: NY
City: New York City

Close the case data. If errors occur you can either add the missing required values or discard the changes.

You have completed the lab on External Data Integration Framework.