

Working with the ERP Integration Service of EMC Documentum Process Services for SAP

Applied Technology

Abstract

EMC® Documentum® Process Services for SAP is a new product that integrates with EMC Documentum Process Services by providing activity templates for common SAP integrations. It also provides a fundamentally new service-oriented interface. The same type of services has always historically been available with a Content Services for SAP agent but was more batch-oriented. With Process Services for SAP, they are exposed as a comprehensive catalog of web services called the ERP Integration Service. These services enable the Content Services for SAP agent's functionality to fully participate in a service-oriented architecture environment, with maximum efficiency and performance. This white paper provides an introduction to the ERP Integration Service and also reviews how to build custom applications using them.

October 2009

Copyright © 2009 EMC Corporation. All rights reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com

All other trademarks used herein are the property of their respective owners.

Part Number h6636

Table of Contents

| | |
|---|-----------|
| Executive summary | 4 |
| Introduction | 4 |
| Audience | 4 |
| Enterprise content service | 4 |
| SOA and web services..... | 4 |
| Architecture and security | 5 |
| Custom applications..... | 5 |
| ERP Integration Service..... | 6 |
| Executing a Process Services for SAP action..... | 6 |
| Executing a external query..... | 6 |
| Sample SAP client applications..... | 7 |
| NetWeaver custom application: Execute service..... | 7 |
| Creating a service destination for ERP services | 7 |
| Creating a Web Dynpro development component | 7 |
| Creating a Web Dynpro component..... | 8 |
| The structure of a Web Dynpro component | 8 |
| ABAP custom application: Link Documentum | 11 |
| Create a package | 11 |
| Create a logical port | 12 |
| Create an ABAP program..... | 12 |
| Step-by-step tutorial | 12 |
| Conclusion | 18 |
| References | 18 |

Executive summary

EMC® Documentum® Process Services for SAP's ERP Integration Service enables both SAP and non-SAP users to access business content directly from their custom applications, eliminating the time and cost associated with searching for, filing, and storing documents. For instance, an accounts payable clerk can instantly see a vendor's contract, invoice, purchase order, and paid check from a single click on an SAP transaction report. Project managers can review standard operating procedures, material safety data sheets, engineering drawings, and specifications from within an SAP material master form.

Users utilizing Microsoft Word and other office products, AutoCAD, e-mail, and any other desktop applications can create and manage digital assets that positively impact corporate processes—from standard operating procedures to material specifications and project plans. Changes to these documents frequently drive supply chain process changes. For example, a change to a material specification requires a change to the SAP material build process. Delays between the release of a specification change and updating the SAP material system can increase scrap materials. With ERP Integration Service, changes to controlled content can automatically initiate the appropriate changes in SAP, decreasing process time, risk, and cost.

And now, the ERP Integration Service has stepped beyond enabling access to the EMC Documentum Content Services for SAP platform via web services and is now providing a complete infrastructure for developing custom applications in the SAP DMS and ECM space. These services provides an interface to web service developers to develop custom applications on an SAP environment.

Introduction

This white paper provides an introduction to the ERP Integration Service and on how to build custom applications using these services.

Audience

This white paper is intended for customers, partners, and EMC support. It is assumed that the audience is familiar with Documentum Content Server, SAP ABAP and NWDI, SOA and web services, and possesses knowledge about EMC storage products.

Enterprise content service

SOA and web services

Service-oriented architecture (SOA) is a paradigm for the realization and maintenance of business processes that span large distributed systems. It is based on three major technical concepts: services, interoperability through an enterprise service bus, and loose coupling.

- A *service* is a piece of self-contained business functionality. The functionality might be simple (storing or retrieving customer data) or complex (a business process for a customer's order). Because services concentrate on the business value of an interface, they bridge the business/IT gap.
- An *enterprise service bus* (ESB) is the infrastructure that enables high interoperability between distributed systems for services. It makes it easier to distribute business processes over multiple systems using different platforms and technologies.
- *Loose coupling* is the concept of reducing system dependencies. Because business processes are distributed over multiple back ends, it is important to minimize the effects of modifications and failures. Otherwise, modifications become too risky, and system failures might break the overall system landscape. Note, however, that there is a price for loose coupling: complexity. Loosely coupled distributed systems are harder to develop, maintain, and debug.

Web services is one possible way of realizing the technical aspects of SOA.

Architecture and security

The ERP Integration Service is a JAX-WS and DFC-based SOA web service. Using these web services, one can write custom applications to achieve the same functionality that was historically part of the agent in the Content Services for SAP.

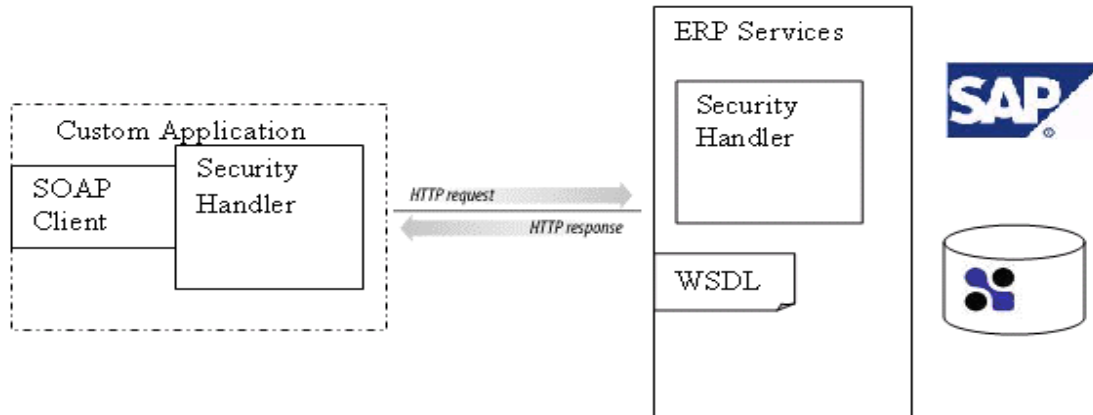


Figure 1. Applications can deploy on any J2EE-enabled application servers.

ERP Integration Service's current release supports UsernameToken-based authentication. In SOAP message-based authentication user credentials are sent via a special SOAP header. For details about UsernameToken, refer to WSS specifications. The WSS UsernameToken authentication profile policy can be configured in Usenametoken-security.xml.

Custom applications

The ERP Integration Service distribution does not contain any client application. To generate a client-side proxy, one of the following can be used:

- JAX-WS tools, as described in JAX-WS documentation
- ABAP development workbench proxy creation, as described in ABAP documentation
- NWDI tools and the Adaptive web service model as mentioned in the SAP Web Dynpro application

Figure 2 shows how custom applications interact with ERP Integration Service.

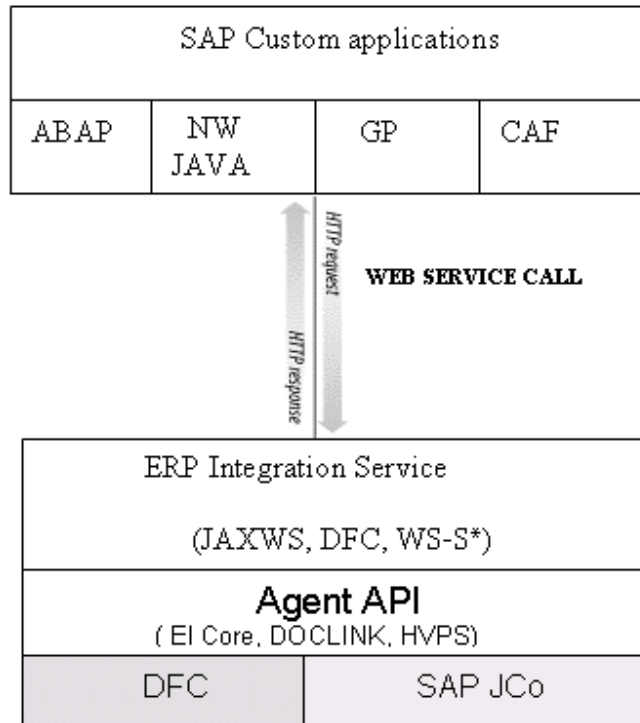


Figure 2. Application interaction within ERP Integration Service

ERP Integration Service

There are two groups of services available in ERP Integration Service. Public services can be consumed by a customer's custom applications. Internal services are suited for Content Services for SAP applications (Content Services for SAP clients, Process Services for SAP, and so on). For a detailed description of service methods and their parameters please, refer to the related Javadoc.

Executing a Process Services for SAP action

Users can execute a preconfigured Process Services for SAP action, which can belong to one of five types:

- Link Documentum
- Link SAP
- Replicate Documentum
- Replicate SAP
- Check DIR Link

To execute a Process Services for SAP action, the user should provide the name of the SAP server configuration, which is configured using the Process Services for SAP WebAdmin.

Executing a external query

Users can execute external (SAP) queries; passing an argument as a type of the SAP objects and configuration mapping.

Sample SAP client applications

NetWeaver custom application: Execute service

This section demonstrates how to create a NetWeaver-based Web Dynpro component. These components can be deployed as a NetWeaver portal application.

Creating a service destination for ERP services

Log in to the NetWeaver administrator using `http://nwserver_ip:port/nwa` and navigate to the **SOA Management > Destination Management** template.

Click the **Create destination** button and fill in the following information:

Destination Type: WSDL

Destination Name: ERP_DESTINATION

URL: <http://localhost:8080/erp/erpintegration?wsdl>

System: JAVA

The Security information can be filled in as shown in Figure 3.

The screenshot shows the 'Security' configuration page for a destination. It is divided into several sections:

- Authentication:** A dropdown menu is set to 'Message Authentication'. Below it, radio buttons are selected for 'User ID/Password (Basic)'. Other options include 'User ID/Password (Digest)', 'X.509 Client Certificate', 'Logon Ticket', and 'SAML Assertion'. A 'Details Off' button is present.
- Details:** A section for 'User ID/Password' with a minus sign icon. It contains three input fields: 'User ID' (containing 'Administrator'), 'Password' (masked with dots), and 'Confirm Password' (masked with dots).
- SSL Server Certificates:** Radio buttons are selected for 'Ignore Server Certificates'. The other option is 'Accept Certificates in Keystore View' with a text box containing 'JCM_SSL_98786'.
- Message Security:** A section with a minus sign icon. It contains a checkbox for 'Use WS-Secure Conversation (Version: February 2005)'. Below this are two sub-sections: 'Outgoing Request' and 'Incoming Response'. Each has two checkboxes: 'Add Encryption' and 'Add Signature' for outgoing, and 'Require Encryption' and 'Require Signature' for incoming. A 'Details' button is at the bottom.

Figure 3. Security information

Creating a Web Dynpro development component

Open the NetWeaver Developer Studio, and select the Web Dynpro perspective by following the menu path of **Window > Open Perspective > Other > Web Dynpro**.

Now select **File > New > Web Dynpro Development Component**.

A window opens for creating new development components. Double-click **MyComponents** [demo.sap.com], type in required fields in the next screen, and click **Finish**.

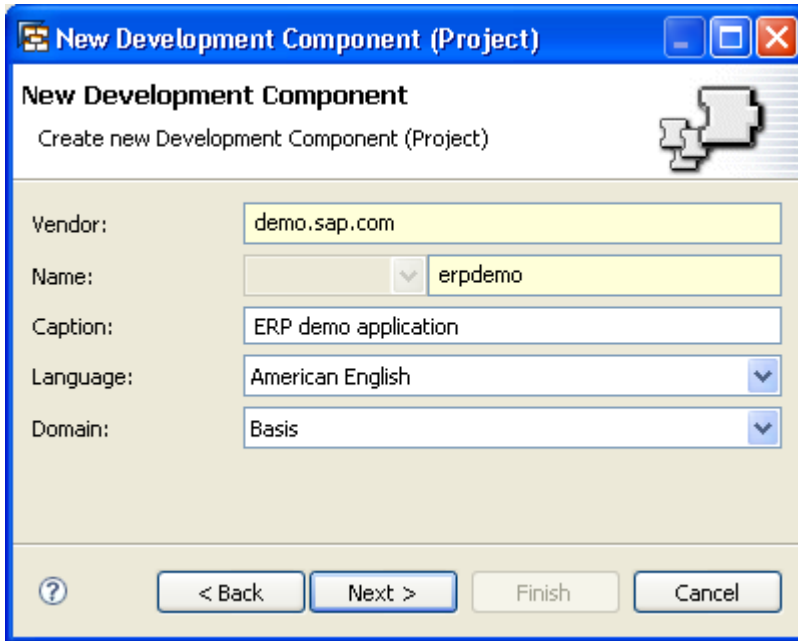


Figure 4. New Development Component dialog box

Creating a Web Dynpro component

You have just created a development component of the type Web Dynpro. It is just a container and does not have any executable on it. To use this container you should create at least one component.

Click the **create component** tool and click on any place in the canvas. As the popup window opens, type the component name “ERPComp” and click **Finish**. On the next screen, add the new views InputView and ResultView and set InputView as the default view.

Now, on your development component canvas you should see a single Web Dynpro component called ERPComp.

The structure of a Web Dynpro component

Double-click on ERPComp and you will see part of the internal structure of the component. You have now drilled down into the component and are looking at the controllers that currently exist within it.

Preparing the Component Controller

In our application, the Component Controller is the central point of control for all functionality and we are asking the user to enter all the parameters required for the web service **executeAction**.

All these user-entered values need to be stored in the Component Controller context. Double-click **ERPComp** and go to the **Context** tab. Right-click on the **Context** node to add a new attribute. Select **manually** and add **repositoryId** as the string type.

Repeat these steps for adding the other parameters serverConfig, userConfig, actionName, and commitFrequency.

Creating a web service model

Click the **Create Model** tool. Click any place in the canvas and select **Adaptive Web Service Model** and click **Next**.

Type the module name **ERP Integration** and module package **com.emc.documentum.erpsservice** and click **Next**. Enter the default metadata destination and execution destination as defined in the section, “Creating a service destination for ERP services.”

Type the metadata destination for the ERP service and click **Next**. In the next screen, type the wsdl location and click **Next**.

You should get a message saying “Model got created” and in Web Dynpro Explorer view, you should see the Model structure.

Mapping a model to component context

Right-click the Component Controller and click **Apply template**.

Select **Service Controller** and click on **Next**. Select the executable model class **Request_ExecuteAction** and click **Next**. Select all the required attributes and click **Next**.

Add the method “executeExecuteAction” for model execution in the controller and click **Finish**.

Create a data link for InputView

Click the **create data link** tool and drag from InputView to the Component Controller. Map the ERPComp attribute to the InputView Context one by one. Click **Finish**.

Adding a template to InputView

Right-click **InputView** and select **Apply template**. Select the **Form** template and click **Next**. Select all the attributes defined in the controller context.

On the next screen correct the order of the input fields in the **InputView** using the arrow mark and click **Finish**.

Double-click **InputView**. Two windows open: a java editor and a view editor. Correct all the labels in the InputView using the Property tab.

Add an “Execute Action” button for navigation, add an action method for the button, and click **Finish**.

Create a data link for ResultView

Follow the same procedure as for InputView and select the **Result** field as a response from the web service call.

Add the Form template to **ResultView** and the item as a display field.

Navigating from one screen to another

When we created our component, a window ERPCompWindow was automatically created by the component wizard. Double-click **ERPCompWindow** and add the link between InputView and ResultView.

Code changes in InputView

Add the following lines to the InputView.java class to call a web service method on the Component Controller.

```
//@@begin onActionExecuteAction(ServerEvent)
    wdThis.wdGetERPCompController().executeExecuteAction();
    wdThis.wdFirePlugOut();
//@@end
```

Code changes in Component Controller

The following code shows how to get an input parameter repositoryId value from InputView in the executeExecuteAction() method on the Component Controller.

```
String repositoryId = (String)
wdContext.getCurrentElement().getAttributeValue("repositoryId");
```

Similarly get the values for serverConfig, userConfig, actionName, and commitFrequency. The following code snippet shows how you can bind and execute the ERPIntegration model.

```
IWDMessageManager manager = wdComponentAPI.getMessageManager();
ERPIntegration model = new ERPIntegration();
Request_ExecuteAction reqExecuteAction= new
Request_ExecuteAction(model);
ExecuteAction executeAction = new ExecuteAction(model);
executeAction.setRepositoryId(repositoryId);
executeAction.setServerConfig(serverConfig);
executeAction.setUserConfig(userConfig);
executeAction.setActionName(actionName);
executeAction.setCommitFrequency(comFreq);
reqExecuteAction.setExecuteAction(executeAction);

wdContext.nodeRequest_ExecuteAction().bind(reqExecuteAction);
wdContext.curentRequest_ExecuteActionElement().modelObject().execute();

wdContext.nodeResponse().invalidate();
```

Making your Web Dynpro component executable

Everything we have done until now has been concerned with the functionality within a Web Dynpro component. However, we need some way of allowing a user access to this component's functionality. This is achieved by creating something called an Application.

First, right-click the **Applications** node in the Web Dynpro Explorer view of the NWDS and select **Create Application**. Enter the required fields and click **Finish**.

Deploy and run your application

Right-click on the name of the application you have just created and select **Deploy new Archive and Run** from the menu. You will then be presented with a pop-up window in which you must enter the deploy user id and password.

Once deployment has finished, the default browser opens and shows the first screen of the application.

Adding a navigation link in the portal

You can add a Web Dynpro application as the pcd link in the portal and add to your defined role.

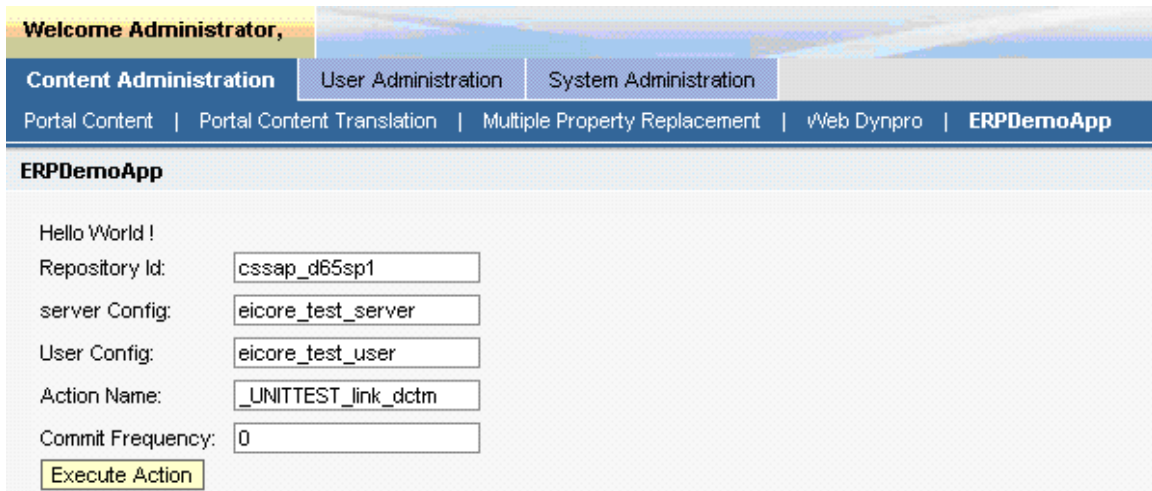


Figure 5. ERP Demo App in a Netweaver portal

ABAP custom application: Link Documentum

As a prerequisite for using this ABAP custom application, an SAP user should have a registered developer available to create package and proxies. They also should have privileges to use transactions like *se80* and *lpconfig*.

Create a package

1. Open an ABAP workbench using the transaction *se80*. Create a package like `ZERP_PACKAGE`. Fill all the required parameters

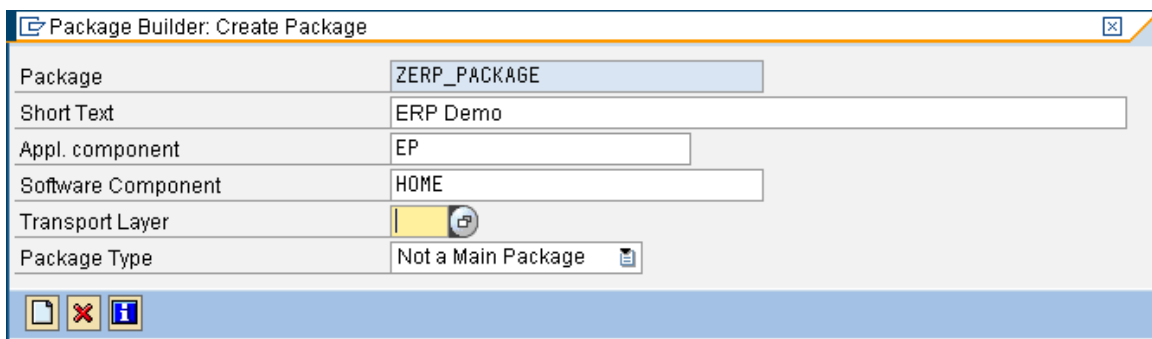


Figure 6. Package Builder: Create Package

2. Right-click `ZERP_PACKAGE` and click the menu **Proxy Object**. Select the **URL/HTTP** destination on the WSDL source popup.
3. Type the WSDL location on the URL text field.
4. Type the package “`ZERP_PACKAGE`” and prefix “`ZERP`” for a web service proxy to be generated. Once you are done creating the web service proxy, click **Save**.
5. Select `ZERPCO_ERPINTEGRATION` and click **Activate**; this should generate all the required proxy classes for the ERP service.

Create a logical port

1. The transaction code for creating a logical port is LPCONFIG. Select a proxy class for ERP Services and type a logical port LP_PORT.
2. Click **Yes** to choose LP_PORT as the default port.
3. On the next screen fill in descriptions like “ERP demo port “. On the **Global setting** tab, select **web service infrastructure** under the Runtime tab. Under the Error tab, select **No Log** and **No trace**. Click **Save**.

Create an ABAP program

To create a ABAP program, on the workbench right-click on **ZERP_PAKAGE** and select a program. On the popup, create a program ZERPPROG of the type “Executable Program”.

You can get the header information

```
ws_header ?= lv_ZERP_ERPINTEGRATION->get_protocol('IF_WSPROTOCOL_WS_HEADER').
```

Here is the code snippet calling the web service:

```
REPORT ZERPPROG.
```

```
DATA: lv_ZERP_ERPINTEGRATION TYPE REF TO ZERPCO_ERPINTEGRATION.  
      CREATE OBJECT lv_ZERP_ERPINTEGRATION  
            EXPORTING LOGICAL_PORT_NAME = 'LP_PORT'.
```

Add additional header information adding UsernameToken tags:

```
DATA: lv_ZERP_REQUEST TYPE ZERPEXECUTE_ACTION1.  
DATA: lv_ZERP_RESPONSE TYPE ZERPEXECUTE_ACTION_RESPONSE1.  
DATA: lv_ZERP_EXCEPTION TYPE ZERPEXCEPTION.
```

Assign the values from the input fields and call the web service method:

```
CALL METHOD lv_ZERP_ERPINTEGRATION->EXECUTE_ACTION  
      EXPORTING  
            INPUT = lv_ZERP_REQUEST  
      IMPORTING  
            OUTPUT = lv_ZERP_RESPONSE.  
      CATCH CX_AI_APPLICATION_FAULT
```

Save the program and click **Activate** to execute the program.

Step-by-step tutorial

This section explains how to use ERP Integration Service for linking a Documentum document to an SAP material object.

1. Create a document “P-503 design specification” using Webtop / da.

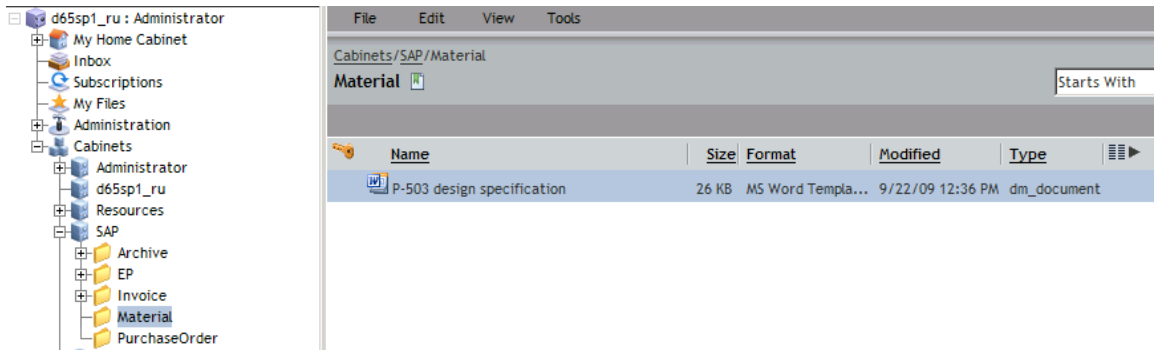


Figure 7. The “P-503 design specification” created in Webtop

2. Configure an SAP connection detail.

Configure the SAP user config object. Here we used “Default User”.

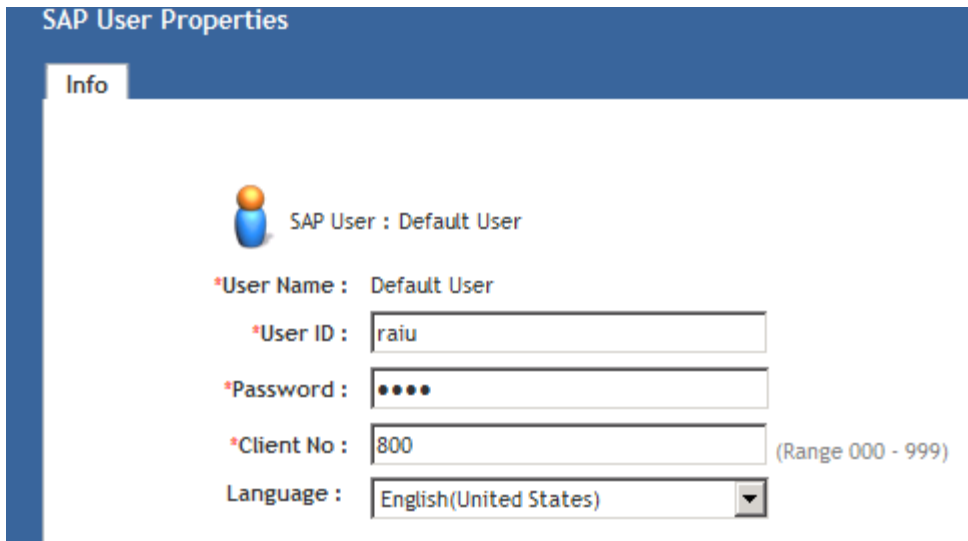



Figure 8. SAP User Properties for the Default User

Configure the SAP Server config object. Here we used “Default Server”.

SAP Server Properties

Info

 SAP Server : Default Server

*Server Name : Default Server

*Host Name/IP :

*System Name : (3 characters)

*System Number : (Range 00 - 99)

Enable load balancing


Figure 9. SAP Server Properties for the Default User

3. Configure PSSAP query and Action objects:

Create a Documentum query that selects the document that needs to be linked with P-503 material.

Documentum Query Properties

Info

 Documentum Query : select P-503 documents

*Query Name : select P-503 documents

DQL Statement :

Select at least attributes r_object_id and object_name!

Results

Items per page

| <u>r_object_id</u> | <u>object_name</u> |
|--------------------|----------------------------|
| 092348788000e2d2 | P-503 design specification |

Figure 10. Documentum Query Properties for “select P-503 documents”

Create a PS SAP Action object associated with the above query object and define a mapping rule with the key assigned as material “P-503”.

Figure 11. Link Documentum Properties for a Link Documentum Demo action

4. Execute the ERP Integration Service method:

The following code line explains how a service method is called. Detailed information on the custom application is available in the “Sample SAP client applications” section.

```
String docbaseName = "d65sp1_ru";
String sapServer = "Default Server";
String sapUser = "Default User";
String actionName = "Link Documentum Demo";
String serviceURL = http://localhost:8080/erp/erpintegration?wsdl;

Service service = Service.create (new URL (serviceURL),
    new QName ("http://erp.service.documentum.emc.com",
"ERPIntegration"));
ERPIntegration servicePort = service.getPort (ERPIntegration.class);

List<String> result =
    servicePort.createLink (docbaseName, sapServer, sapUser,
        actionName, 0);
logger.info ("Create Link returns: " + result);
```

5. Document consumption is next. This step helps the user to verify the created DIR links and view the linked document in the Documentum repository.

Use transaction mm03 to display Material Master Records. Enter P-503 to navigate to the Material used:

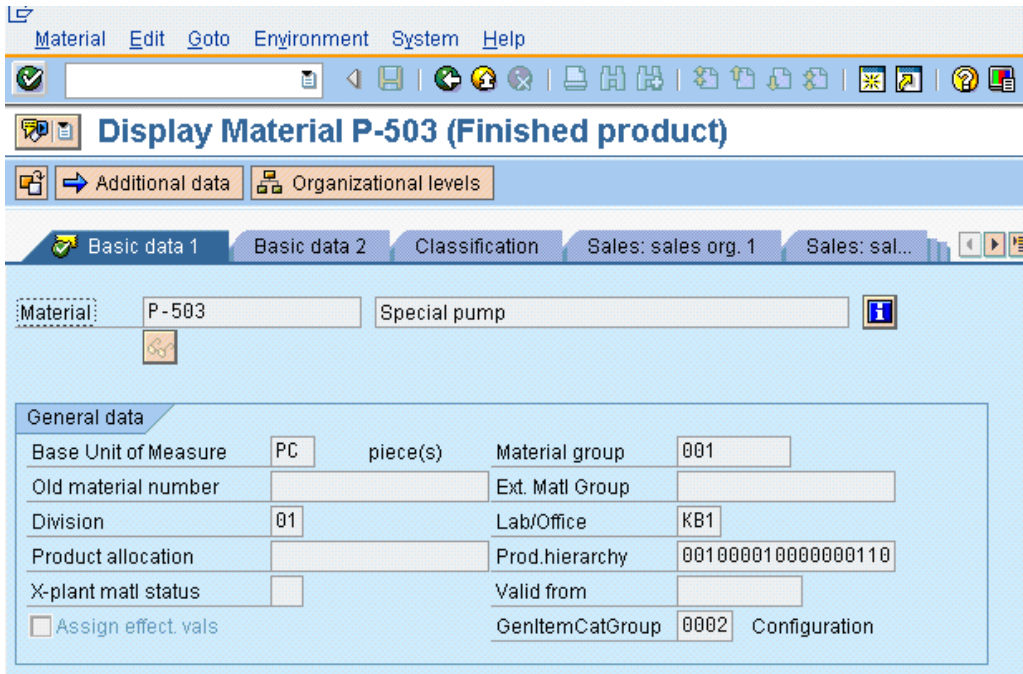


Figure 12. Material Master Record for Material P-503

Click **Additional data** as shown in Figure 12 and click the **Document data** tab to view the added DIR as shown in Figure 13.

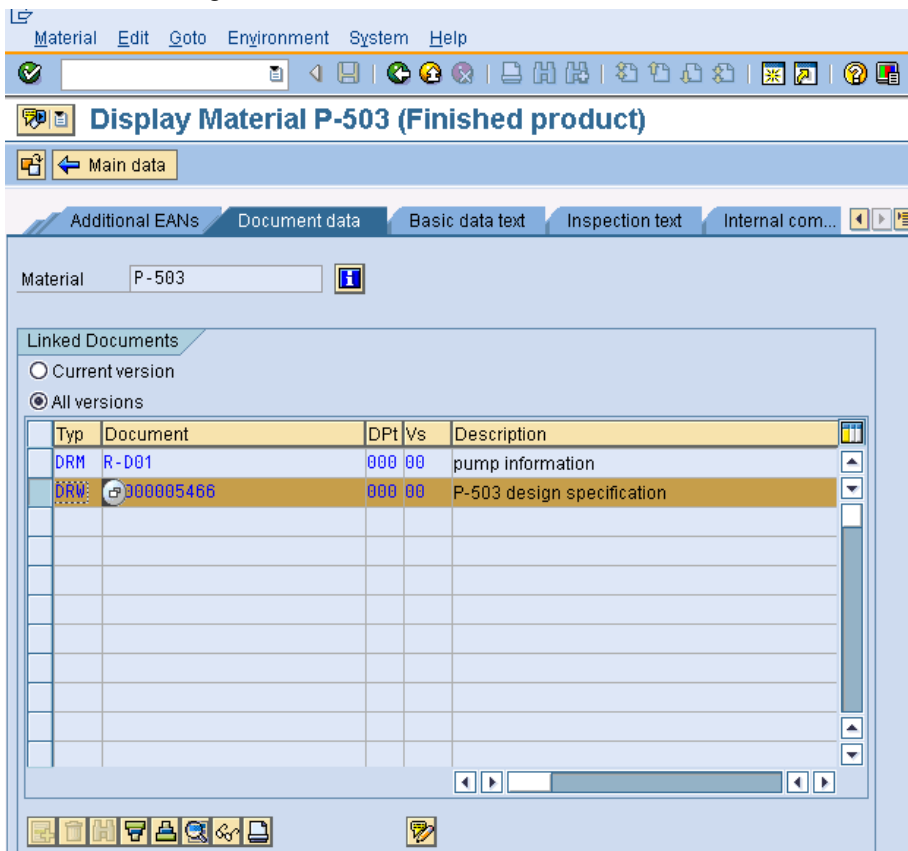
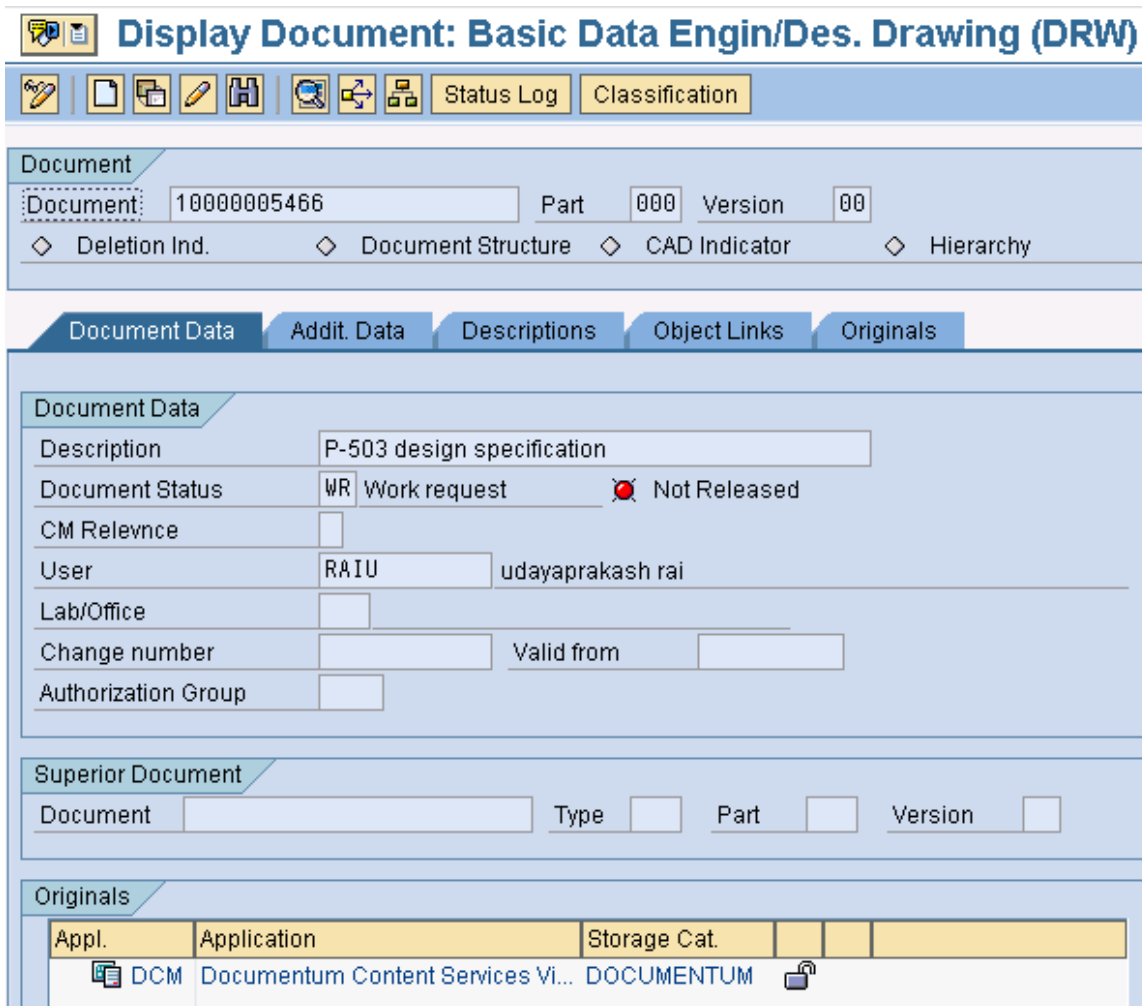


Figure 13. The DIR created for P-503

Click the magnifying glass  in Figure 13 to navigate the DIR.



Display Document: Basic Data Engin/Des. Drawing (DRW)

Document

Document: 10000005466 Part 000 Version 00

◇ Deletion Ind. ◇ Document Structure ◇ CAD Indicator ◇ Hierarchy

Document Data Addit. Data Descriptions Object Links Originals

Document Data

Description P-503 design specification

Document Status WR Work request Not Released

CM Relevnce

User RAIU udayaprakash rai

Lab/Office

Change number Valid from

Authorization Group

Superior Document

Document Type Part Version

Originals




| Appl. | Application | Storage Cat. | | |
|---|-----------------------------------|--------------|---|--|
|  DCM | Documentum Content Services Vi... | DOCUMENTUM |  | |

Figure 14. Display Document screen for the linked document

Click the glasses  in Figure 13 to view the document directly.

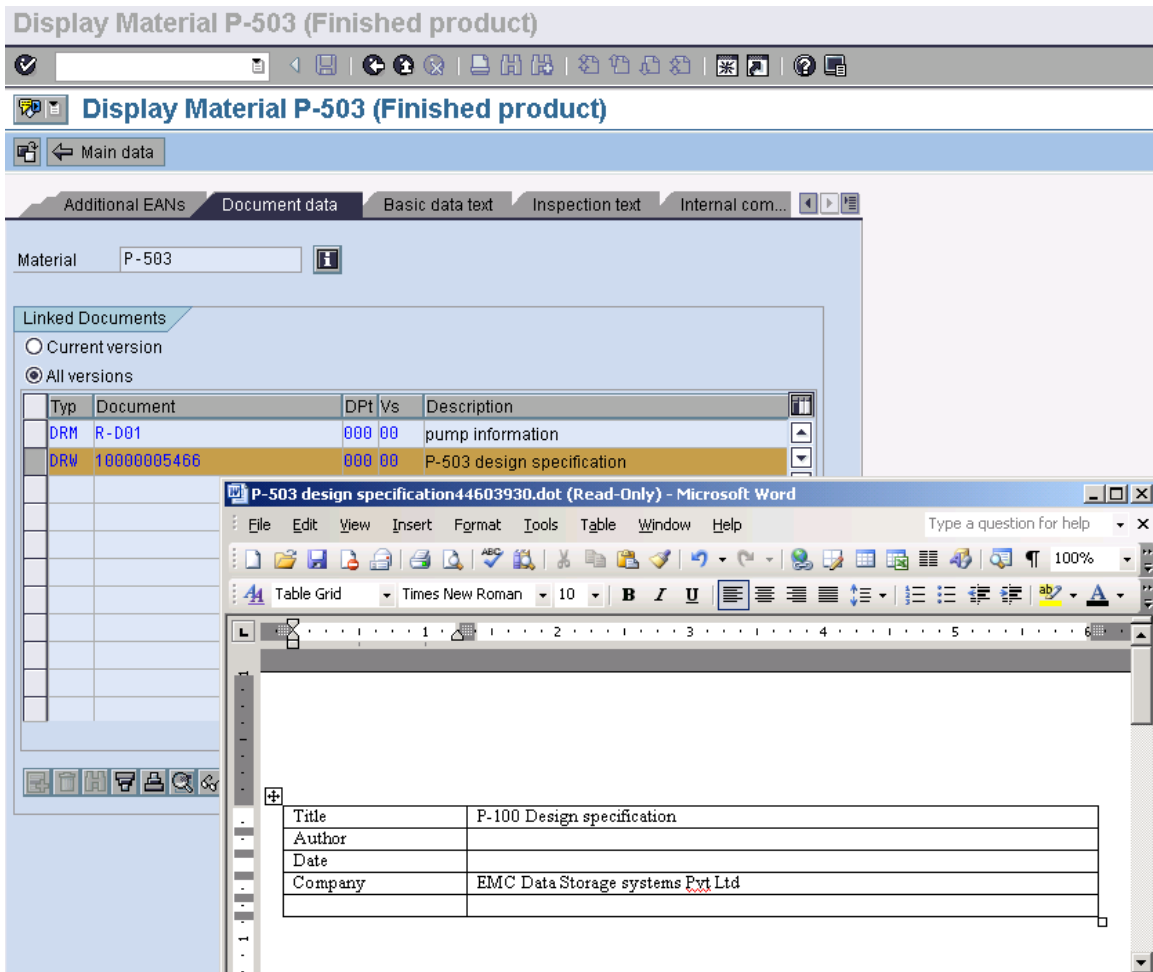


Figure 15. Linked document to the P-503 material

Conclusion

EMC Documentum Process Services for SAP now exposes ERP Integration Service—a set of web services enabling the next generation of SAP integrated content management applications. It is now providing a complete infrastructure for developing custom applications in the SAP DMS and ECM space. These services provide an interface to web services developers to develop custom applications on the SAP environment. The result is a degree of flexibility that will reduce traditional limitations and resource constraints associated with developing custom applications that are unique to an enterprise's business environment.

Through the ERP Integration Service the developer is equipped to leverage functionality previously available in the Content Services for SAP agent, for a wider range of business processes and services. Whether our customers are embracing service orientation as an architectural approach or are simply moving their enterprise toward the power of web services capabilities, SOA-based ERP Integration Service is a powerful choice.

References

For details about Documentum suite of products, go to the Documentum Family page on EMC.com: <http://www.emc.com/products/family/documentum-family.htm>