Integrating Lotus Domino Workflows in SAP Enterprise Portal 6.0

Applies to:
SAP Enterprise Portal 6.0, IBM Lotus Domino 6.5.3, SAP R/3 4.6C

Article Summary
This article presents the technical details required for the integration of the SAP Enterprise Portal 6.0 with Lotus Domino and SAP R/3 4.6C. It also outlines the required software components and classes with necessary code snippets.

This exercise, in which the workflow management is delegated to Lotus Domino, was demonstrated to one of our customers.

The solution described is one of the ways of approaching such integration. This article attempts to provide the technical tips that would eliminate the time and effort to invest in similar business requirements.

By: Madhusudhan B / Umasankar S
Title: Associate Consultant / Consultant (Wipro Technologies, Bangalore, India)
Reviewed By: Kartik Iyengar, Group Lead,
NetWeaver Competence Group (NWCG),
Wipro Technologies, Bangalore, India

Date: 24 Feb 2005

Table of Contents
Applies to: .................................................................1
Article Summary ..........................................................1
Table of Contents ..........................................................1
Introduction .................................................................2
A Brief Introduction to Lotus Domino ................................2
Business Case ...............................................................4
System Architecture .....................................................5
Collaborating Components ...........................................6
SAP EP/SAP Web AS: .....................................................6
Lotus Domino: ............................................................6
Technical Details ..........................................................6
  Step 1: Create the Web Dynpro Required .........................6
  Step 2: Code the Model Object to Hold the Purchase Requisition Values ..................................7
  Step 3: Code the Servlet Required to Intercept the Model Object Passed from Web Dynpro. 7
  Step 4: Code Sample for the Agent to End Mail and Invoke Subsequent Agents ..................8
  Step 5: Code snippet for the agent to create the Purchase Order in SAP R/3 .........................9
  Step 6: Display the Progress in a Separate Browser Window ...........................................11
Technical Flow – Create Purchase Order from Purchase Requisition .................................12
Summary and Conclusion ...............................................13
Disclaimer & Liability Notice ...........................................13
Author Bio .................................................................14
Introduction

In our company, we came across a specific requirement from one of our customers that we need to demonstrate the creation of a purchase order in SAP R/3 4.6C with the approval workflow through Lotus Domino. For this, we had taken a sample business case of creating a purchase requisition from SAP EP/Web Dynpro and handing it over to Lotus Domino for the approval workflow. Once the cycle is successfully done, then we create a purchase order in SAP R/3 4.6C using the Lotus Enterprise Integrator (LEI) connector for Domino.

We had to overcome the challenge of not having experience with Lotus Domino. Specifically, we had to focus our efforts on:

- Integration of Web Dynpro with Domino
- Handling the approval cycle within Domino
- Connection configuration in Domino for the SAP R/3 4.6C
- Making both synchronous and asynchronous calls work

In this tutorial, we will focus on the above issues and resolution measures that provide an idea of how easily integration can be achieved, so that similar requirements can be easily solved without investing much time and effort.

We start with a brief introduction of Lotus Domino in the following section.

A Brief Introduction to Lotus Domino

Lotus Domino, a server software from IBM, combines enterprise-class messaging and calendar/scheduling capabilities with a robust platform for collaborative applications on a variety of operating systems. This server comes in three versions:

- Domino Messaging Server- messaging only
- Domino Utility Server- applications only
- Domino Enterprise Server- both messaging and applications

Lotus Notes is the workstation (client) software that provides an integrated collaborative environment. This allows users to access the messaging and collaboration capabilities of the Lotus Domino server. Other options available for client are Domino Web Access, Domino webmail, etc.
Domino Designer is application development software that allows application developers and web site designers to create secure, collaborative applications that can be accessed via Lotus Notes or web browsers. Developers can easily create core business solutions based on forms, views, pages, framesets, integrated instant messaging, XML, Java, JavaScript, etc.

Lotus Administrator is used for the administration activities of the Domino Server. For this exercise we use a combination of Domino, Notes, Designer, and Administrator for the workflow management.

The following section details the business case taken for this example.
Business Case

Let us examine the system architecture for the subject scenario in the next section.
System Architecture

The following system set up diagram shows all the software and hardware components involved in the integration.
Integrating Lotus Domino Workflows in SAP Enterprise Portal 6.0

Collaborating Components

SAP EP/SAP Web AS:

- Web Dynpro to create the Purchase Requisition
- Web Dynpro to view the Purchase Requisition and submit for Approval on Lotus Domino
- Model Object to hold the PR values

Lotus Domino:

- Domino configuration for the LEI connector for R/3
- Servlet to process the Model Object passed from Web Dynpro
- Agent to receive mails and take actions
- Agent to create PO in R/3

Technical Details

With these components in place, the following steps describe the associated development and integration. Code examples are provided wherever necessary.

Note: The code samples provided here are not complete.

Step 1: Create the Web Dynpro Required

The following screen shot shows the purchase requisition (PR) created and submitted for the approval cycle.
Step 2: Code the Model Object to Hold the Purchase Requisition Values

The following code shows the model object code:

```java
public class PRObject implements java.io.Serializable
{
    String reqItem;
    String reqNo ;
    BigDecimal quantity;
    String material;
    String unit;

    public void setVendor(String vendor)
    {
        this.vendor = vendor;
    }

    public String getVendor()
    {
        return vendor;
    }

    public void setStorageLocation(String storageLocation)
    {
        this.storageLocation = storageLocation;
    }
}
```

Step 3: Code the Servlet Required to Intercept the Model Object Passed from Web Dynpro

Here, the important code part of how the Java object is passed from one servlet container to the domino container is shown.

Following is the code in the Web Dynpro application that opens a URLConnection with the servlet deployed on Domino Server 6.5.3, and sends the purchase requisition details to the servlet for the approval process.
URL url = new URL("http://<Hostname>/servlet/CreatePurchaseOrderServlet");
URLConnection urlConnection = url.openConnection();
urlConnection.setRequestProperty("Content-Type", "application/octet-stream");

urlConnection.setDoInput(true);
urlConnection.setDoOutput(true);

urlConnection.setUseCaches (false);
urlConnection.setDefaultUseCaches (false);

ObjectOutputStream outStream = new ObjectOutputStream(urlConnection.getOutputStream());

outStream.writeObject(PRObject);
outStream.flush();
outStream.close();

Following is the code snippet in the servlet that opens an InputStream on a URL Connection with the Web Dynpro application. It reads the purchase requisition details that are persisted in an object and sent from the Web Dynpro application.

ObjectInputStream inStream = new ObjectInputStream (req.getInputStream ());
PRObject prObject = (PRObject)inStream.readObject ();
inStream.close();

Step 4: Code Sample for the Agent to End Mail and Invoke Subsequent Agents

Following is the code snippet of a control agent who checks for the approval:

if(document != null)
{
    sentBy = document.getItemValueString("Principal");
    sentTo = document.getItemValueString("From");
subject = document.getItemValueString("Subject").toLowerCase();

if(sentBy.indexOf("First Supervisor") !== -1)
{
    if(subject.indexOf("approved") !== -1)
    {
        Agent agent = database.getAgent("MailingAgent");
        agent.runOnServer();
    }
}
else if(sentBy.indexOf("Second Supervisor") !== -1)
{
    if(subject.indexOf("approved") !== -1)
    {
        NotesThread.sinitThread();
        Session sess = NotesFactory.createSession();
        AgentContext agentContext = sess.getAgentContext();
        Database db = session.getDatabase("6SCS-110409","sapexamp.nsf");
        Agent mailingAgent = db.getAgent("MailingAgent");

        mailingAgent.runOnServer();
    }
}

Step 5: Code snippet for the agent to create the Purchase Order in SAP R/3

Set sapsession = New LCConnection("sap")
sapsession.Userid = “userid”
sapsession.Password = “passord”
sapsession.Client= 100
sapsession.Destination= SAP Host
sapsession.SystemNo= 00
sapsession.Language= “EN”
sapsession.Server= SAP Host
sapsession.Connect
Dim itmField As NotesItem
Dim itmValue As NotesItem

Dim index As Long
Dim intIndex As Integer
Dim itemCount As Integer

Dim inputlist As New lcfieldlist
Dim outputlist As New lcfieldlist

sapsession.Procedure = "ME21"
sapsession.ModuleType = 1
sapsession.Metadata = "*"

Dim wrkstring As String
wrkstring = POString
sapsession.ScreenFields = wrkstring
sapsession.MapByName=True

Call sapsession.Call(inputlist,1,outputlist)

If (sapsession.Fetch (outputlist) > 0) Then
    Set field = outputlist.lookup("EXPORTSMESSGMSGTX",index)
    Print "Result : " & field.text(0)
Step 6: Display the Progress in a Separate Browser Window

Lotus Workflow Management

- Purchase Requisition Details Received from Enterprise Portal
- Purchase Requisition Stored into Lotus Domino Database
- Mail Sent to First Level of Approval
- Received the Confirmation and Mail Sent to Second Level of Approval
- Purchase Requisition Details sent to B/S
- Standard PO created under the number 4500000096
1. Model object is passed to Domino servlet.
2. Servlet saves the data in Domino DB (PR data).
3. Invokes a control agent which in turn invokes the mailing agent.
4. Control agent invokes the mailing agent to send mail to primary approver.
5. Primary approver after approving (or rejecting) sends mail to all concerned.
6. Control agent reads the mail and if accepted, invokes mailing agent this time to send mail to secondary approver.
7. Secondary approver after approving (or rejecting) sends mail to all concerned.
8. Control agent reads the mail and if accepted, invokes the CreatePO agent.
9. CreatePO agent sends the data through the LEI connector to insert into SAP R/3.
Summary and Conclusion

In this article we have seen the details involved with the integration of SAP NetWeaver with Domino and SAP R/3 4.6C. We have also tried to demonstrate the same with sufficient screen shots and code samples wherever required. We hope this tutorial gives sufficient information to take up similar integration scenarios without investing much time and effort.

We invite your feedback comments on this article:
umasankar.subramanian@wipro.com
Madhusudhan.buragapally@wipro.com

Disclaimer & Liability Notice

This document may discuss sample coding, which does not include official interfaces and therefore is not supported. 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 of the code and methods suggested here, and anyone using these methods, is doing it under his/her own responsibility.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of the technical article, including any liability resulting from incompatibility between the content of the technical article and the materials and services offered by SAP. You agree that you will not hold SAP responsible or liable with respect to the content of the Technical Article or seek to do so.
Author Bio

Madhusudhan B has over four years of experience in Java/J2EE technologies and is currently working as an SAP NetWeaver consultant at Wipro Technologies and as a certified SAP XI 3.0 development consultant. He has been involved in SAP NetWeaver for the past year, focusing on SAP Enterprise Portal 6.0 and SAP XI 3.0.

Umasankar S is working as a consultant in SAP NetWeaver Competence Group (NWCG) at Wipro Technologies, Bangalore. He is predominantly worked on designing and executing J2EE projects in the shipping domain, and is essentially a mechanical engineer having worked in the manufacturing industry also. Currently, his area of focus involves architecting EAI applications (SAP NetWeaver) and application frameworks.