Adeptia Suite

User Training

(2-Day Session)

443 North Clark Street,
Suite 350,
Chicago, IL 60654
www.adeptia.com

ADEPTIA

Goal

- Familiarize the audience with the capabilities of the Adeptia Suite
- Train the audience to utilize the Adeptia Suite to solve complex business problems
- Pre-requisites
  - Familiarity with Java
  - Familiarity with Application Integration concepts
  - Familiarity with Business Process Automation concepts
  - Familiarity with Service Oriented Architecture (SOA)
  - Reading of “Understanding Adeptia Suite”

Training schedule

Day 1

Morning session
9am to 12pm, 10 min break at 10:30am

- Introduction
  - Overview of Adeptia Suite
  - Architecture
  - Project approach
- Process Modeler & Simulation (BPM)
  - Modeler
  - Simulator
- Process Designer
- BPM Exercise – Purchase Requisition Workflow (BPM)
  - Web Forms
  - Workflow

Afternoon session
1pm to 4:30pm, 10 min break @ 2:20pm and 3:20pm

- Integration/ETL Exercises:
  - Employee Benefits
  - Purchase Order
  - Inventory Items
  - Order Fulfillment
- Services: Description and Exercises
  - Sources
  - Targets
  - Data Dictionary
  - Schema
  - Mapping
  - Web Services
  - Workflow
- Events & Triggers

Day 2

Morning session
9am to 12pm, 10 min break at 10:30am

- Advanced Mapping & Process Designer functions
  - Process Designer
    - Decision nodes
    - Context Variables
    - Sub Processes
  - Mapping
    - Math, String and Conditional functions
    - Database lookups
    - Context variables
    - Multiple Source and Target mapping
- Execute flow
  - Trigger
  - Logs
  - Reports

Afternoon session
1pm to 4pm, 10 min break @ 2:30pm

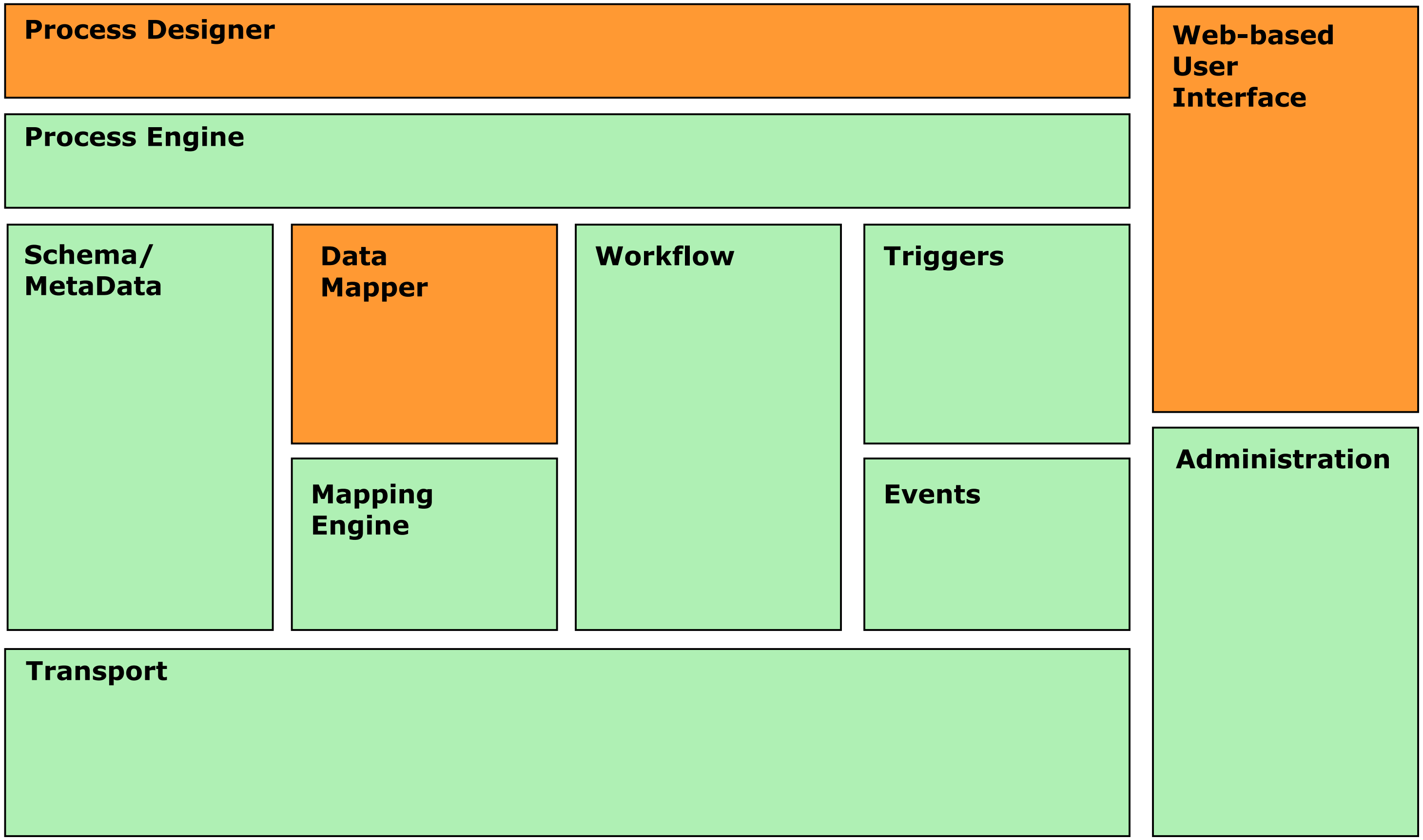
- Managing Solutions
  - My Tasks and Task History
  - Solution and Performance Monitoring
  - Logs
- Administration
  - User management
  - Configuration and Properties
  - System monitoring
  - Maintenance activities

Introduction

Overview: Key feature areas

- Manage
  - Document business processes
  - Share and collaborate between business users and IT
  - Manage and track workflow tasks
- Automate
  - Create individual services and activities
  - Implement execution of process flows
- Optimize
  - Track and monitor
  - Analyze logs and reports to identify bottlenecks
  - Enhance and improve performance
- Administer
  - Allows management of Groups and Users
  - Configure and manage application parameters

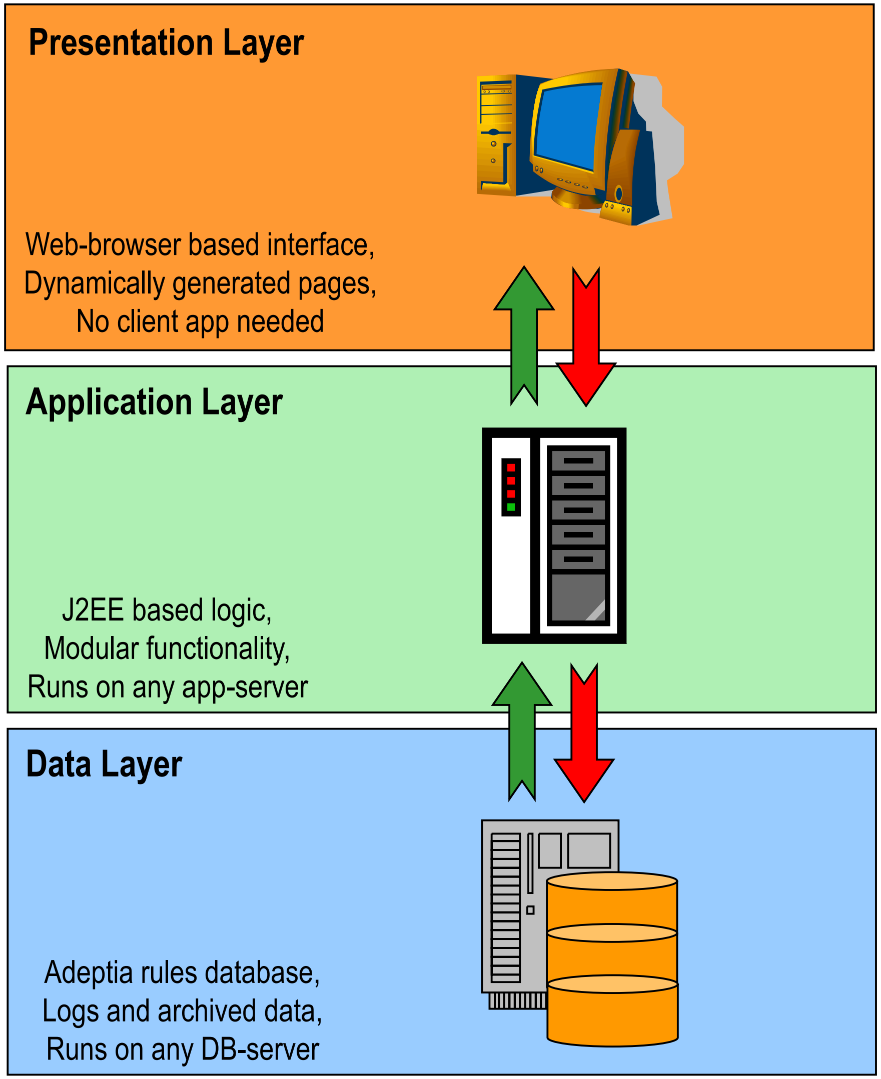
Functional Modules



🟧 **User-interface, design time**

🟦 **Back-end, run time**

Application Architecture



Presentation Layer

Web-browser based interface,
Dynamically generated pages,
No client app needed

Application Layer

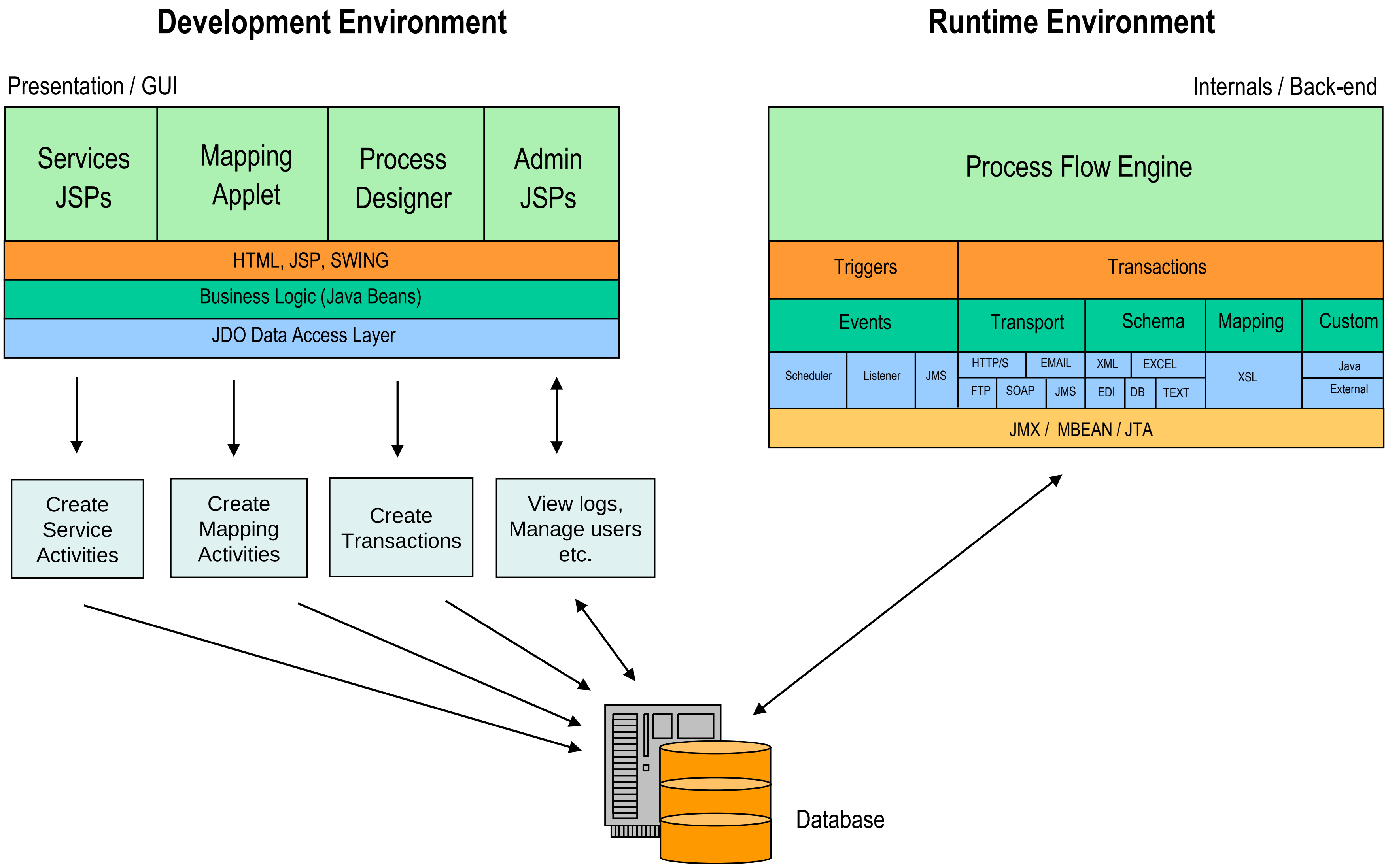
J2EE based logic,
Modular functionality,
Runs on any app-server

Data Layer

Adeptia rules database,
Logs and archived data,
Runs on any DB-server

- Java based application
- N-Tier application architecture
- Utilizes XML, XSLT, Java (JMX, JDO, JCA, JMS etc.), Swing and Web services
- All business rules, design rules, activities and process flows are archived in a database
- Internal data flows, rules and process are in XML format

Architecture: Module stack



Development Environment

Presentation / GUI

Services JSPs | Mapping Applet | Process Designer | Admin JSPs

HTML, JSP, SWING

Business Logic (Java Beans)

JDO Data Access Layer

Create Service Activities

Create Mapping Activities

Create Transactions

View logs, Manage users etc.

Runtime Environment

Internals / Back-end

Process Flow Engine

Triggers | Transactions

Events | Transport | Schema | Mapping | Custom

Scheduler | Listener | JMS | HTTP/S | EMAIL | FTP | SOAP | JMS | XML | EXCEL | EDI | DB | TEXT | XSL | Java | External

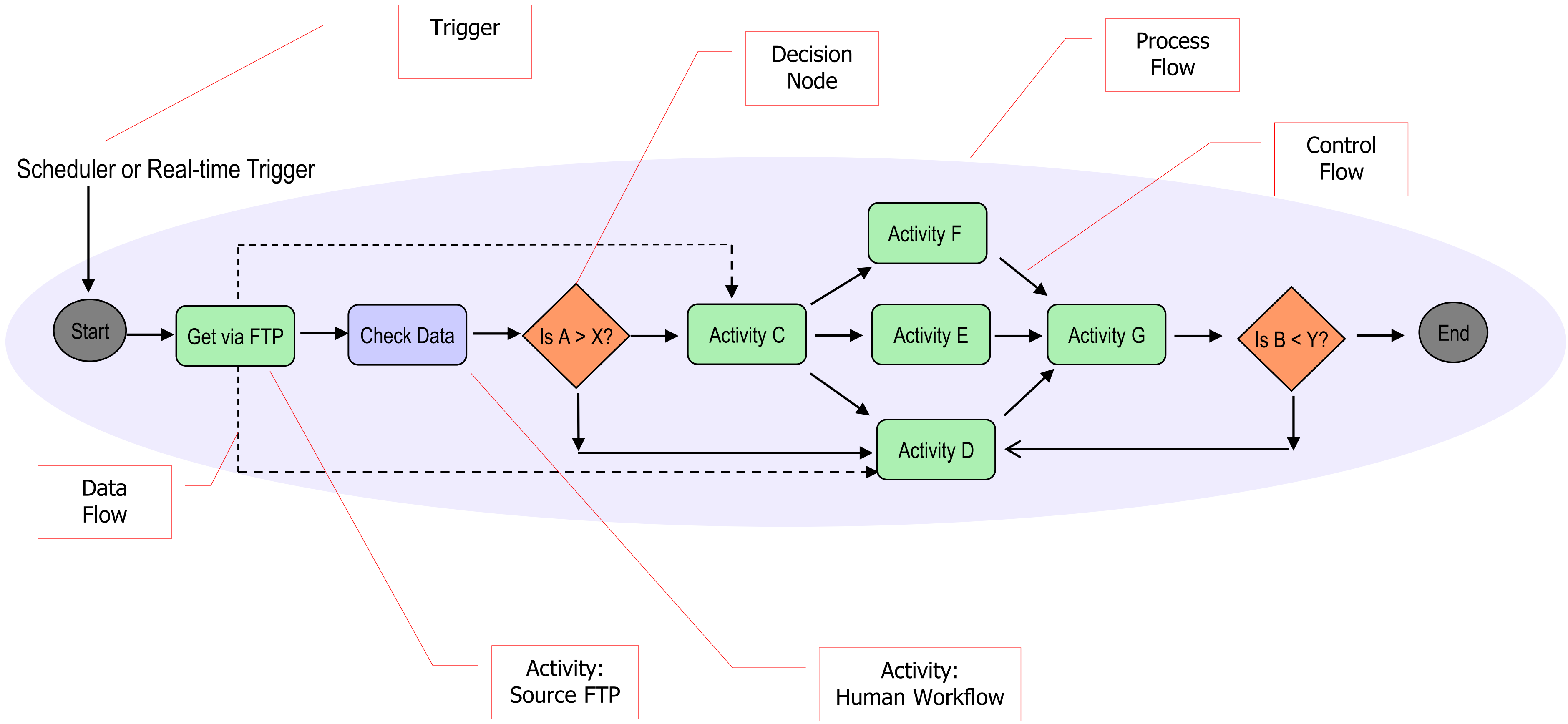
JMX / MBEAN / JTA

Database

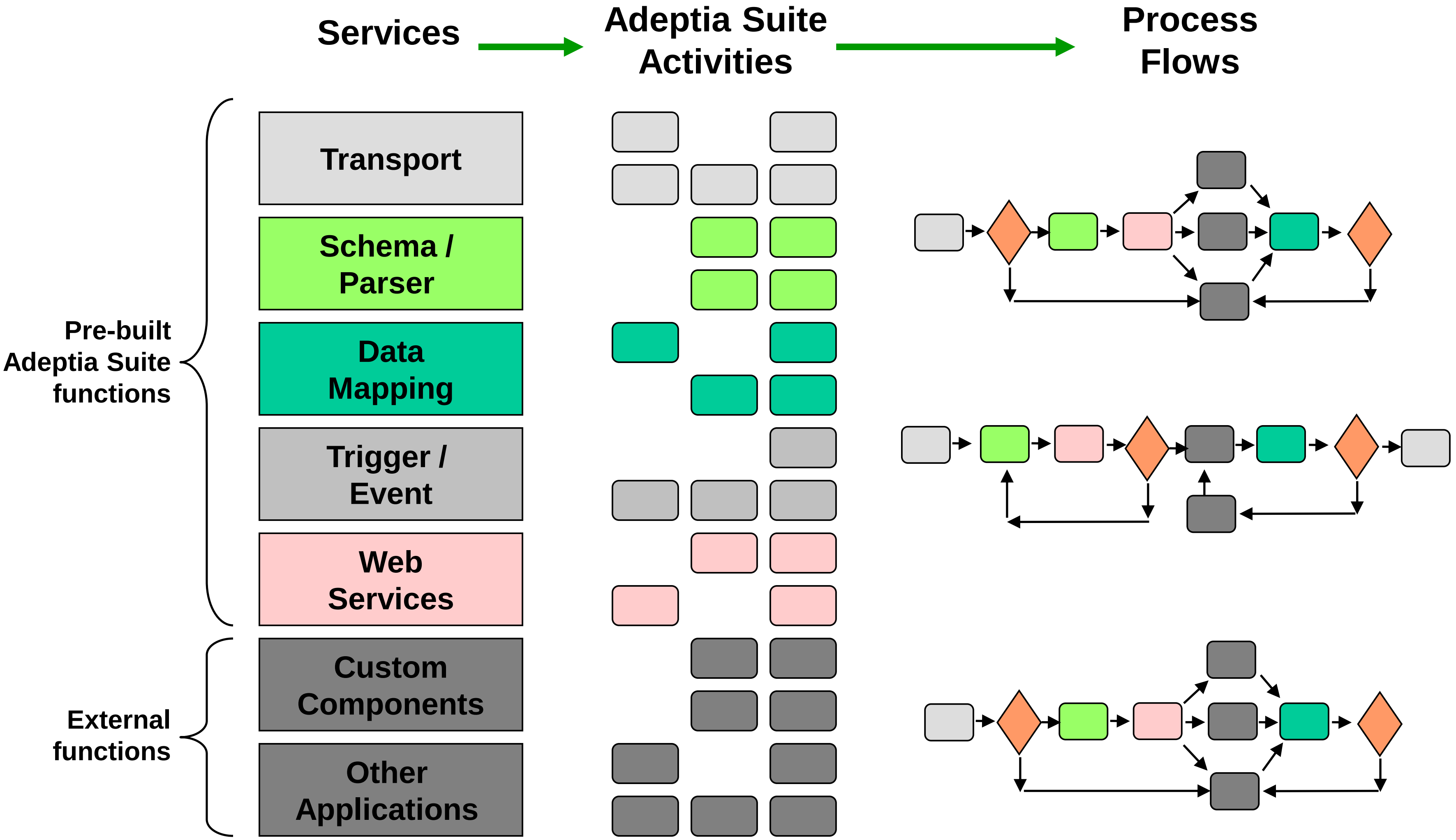
Key concepts

- Developers:
  - These are IT staff or consultants who use the Adeptia Suite to design and implement process flows
- Business Users:
  - These are end-users who benefit from the automation of business processes and interact with flows using workflow tasks
- Design time:
  - This is the phase during which developers implement flows
- Run time:
  - This is the phase when the process flows that have been implemented are run in production environment
- Service:
  - This is a type of feature or specific business task or action
- Activity:
  - This is an instance of a service which is actually executed in a flow
- Process Flow:
  - This is a set of activities that combine together to create a business process

Key concepts



Services based approach



Services → Adeptia Suite Activities → Process Flows

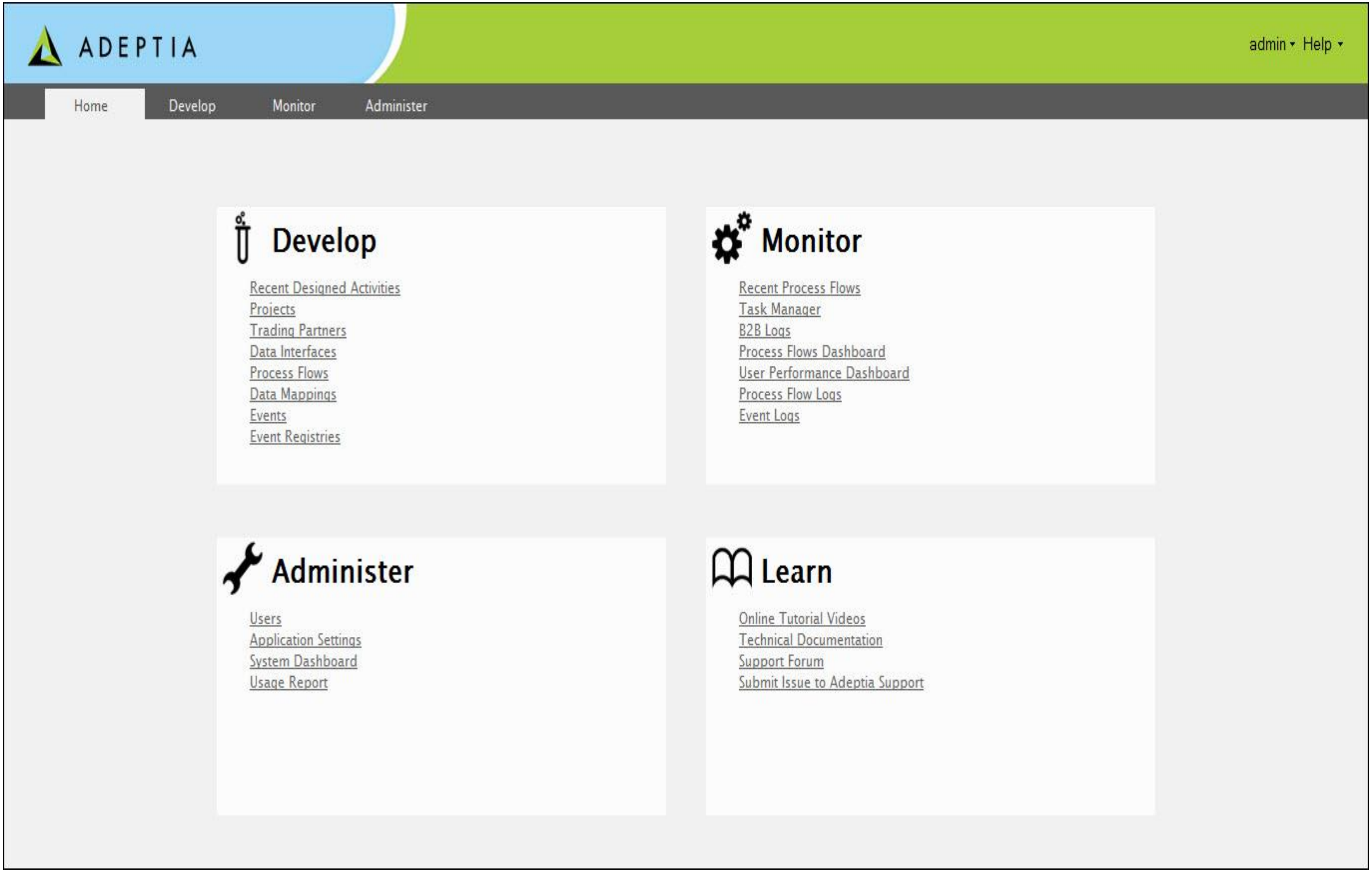
Pre-built Adeptia Suite functions

- Transport
- Schema / Parser
- Data Mapping
- Trigger / Event
- Web Services

External functions

- Custom Components
- Other Applications

Quick product tour



Develop

- Recent Designed Activities
- Projects
- Trading Partners
- Data Interfaces
- Process Flows
- Data Mappings
- Events
- Event Registries

Monitor

- Recent Process Flows
- Task Manager
- B2B Logs
- Process Flows Dashboard
- User Performance Dashboard
- Process Flow Logs
- Event Logs

Administer

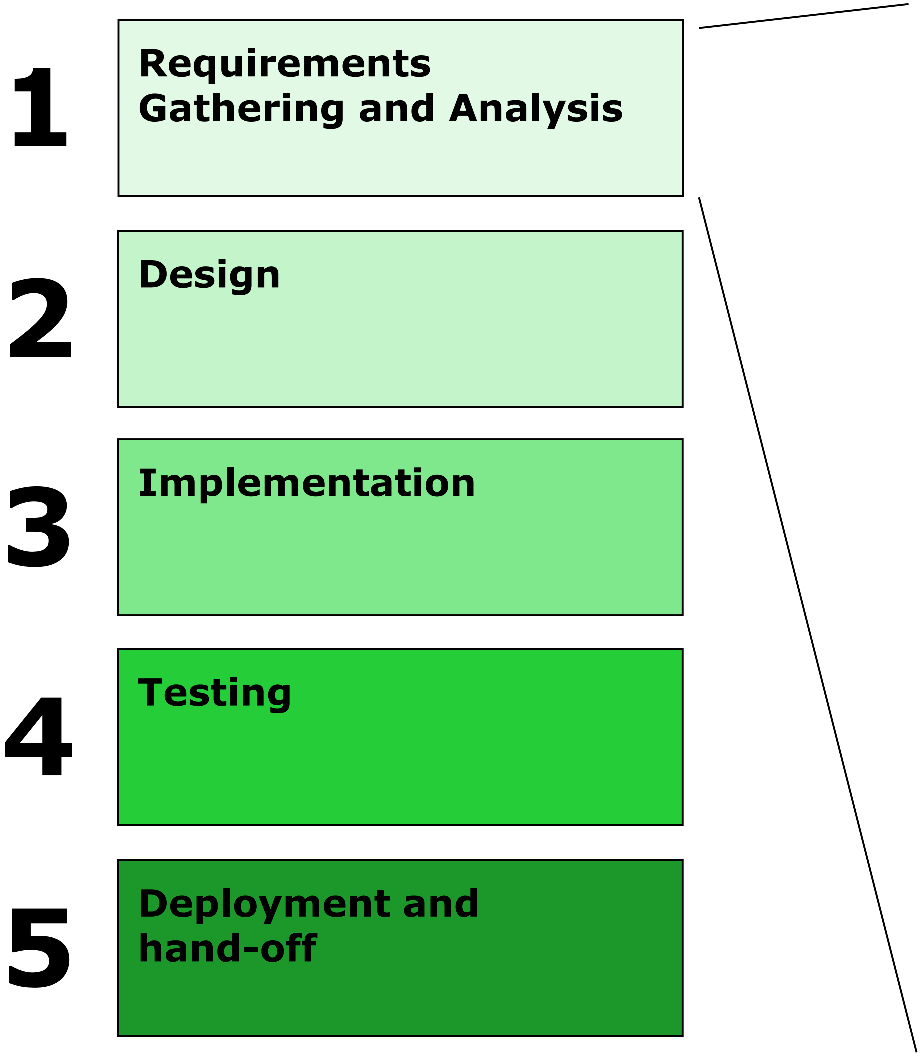
- Users
- Application Settings
- System Dashboard
- Usage Report

Learn

- Online Tutorial Videos
- Technical Documentation
- Support Forum
- Submit Issue to Adeptia Support

Typical Project Approach

Project steps



1 Requirements Gathering and Analysis

2 Design

3 Implementation

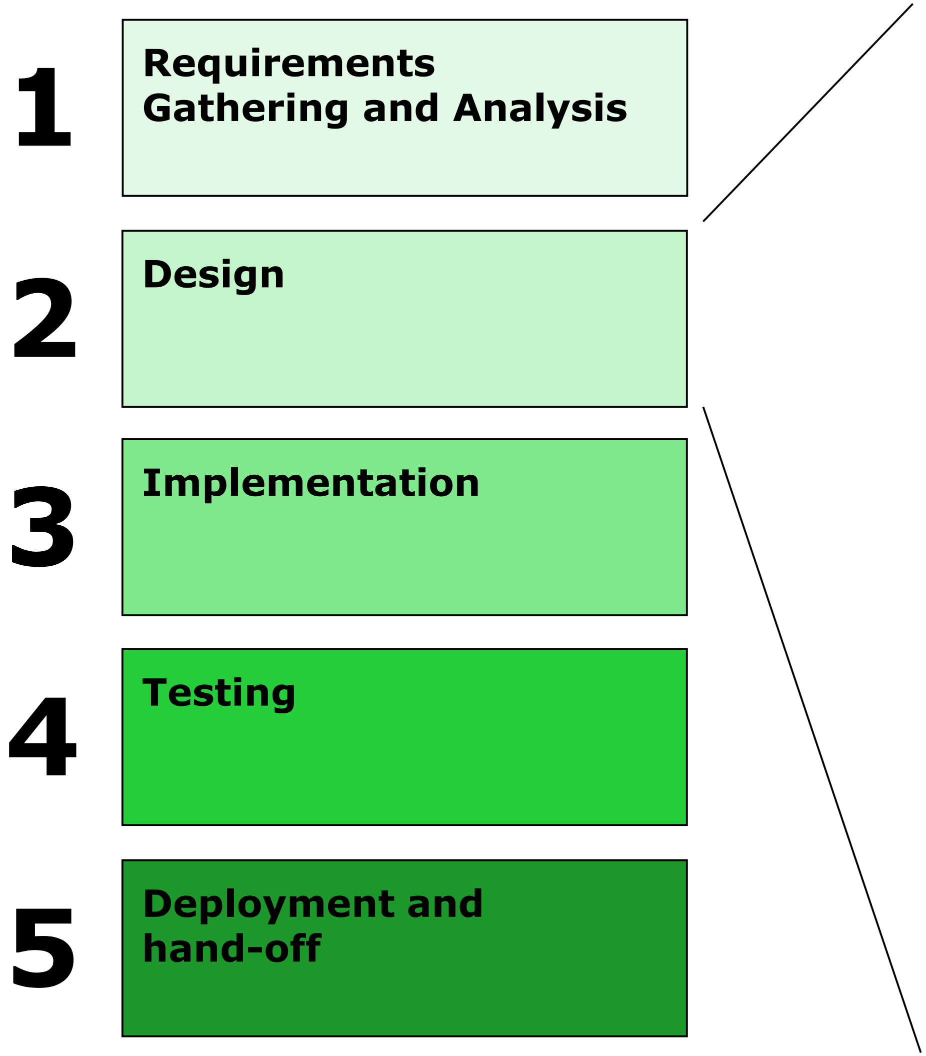
4 Testing

5 Deployment and hand-off

Requirement Gathering and Analysis

- Business requirements
  - Business processes
  - Business rules
  - Business users
  - Reporting
- Technical requirements
  - Concurrent execution of flows
  - Applications and their interfaces
  - Data volume
  - Security

Project steps



1 **Requirements Gathering and Analysis**

2 **Design**

3 **Implementation**

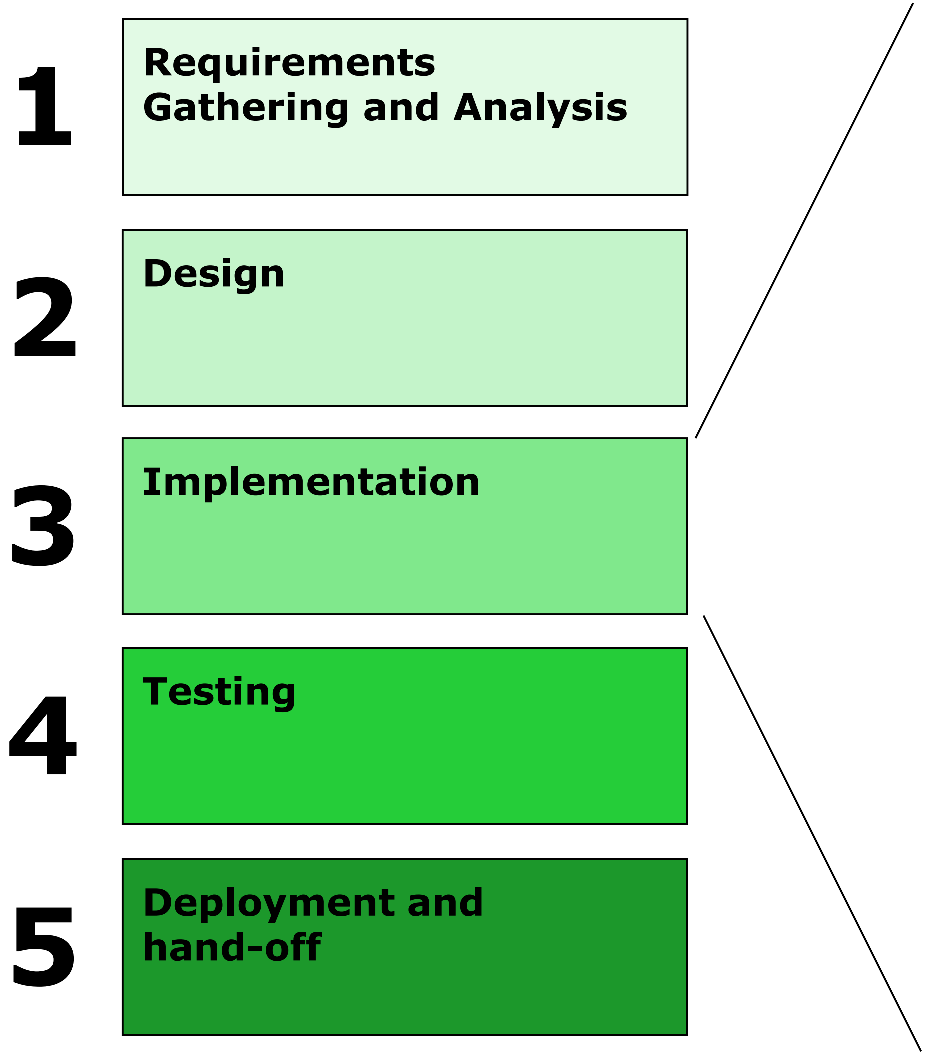
4 **Testing**

5 **Deployment and hand-off**

Design

- Document business process
  - High-level view of flow
  - List of steps and tasks
- Map business process to Adeptia flow
  - Flow activities and steps
  - Decision nodes
  - Workflow steps
  - Exception Handling
  - Execution mode (Manual, Event, Scheduled, Concurrent etc.)
  - Business users and Security
- Design Activities and Integration
  - Document activity parameters
  - Document data schema and formats
  - Specify data mapping rules
  - Data interfaces and protocols
  - Database drivers
  - Timers, Events and Triggers
  - Human workflow details
  - Application interface formats

Project steps



1 **Requirements Gathering and Analysis**

2 **Design**

3 **Implementation**

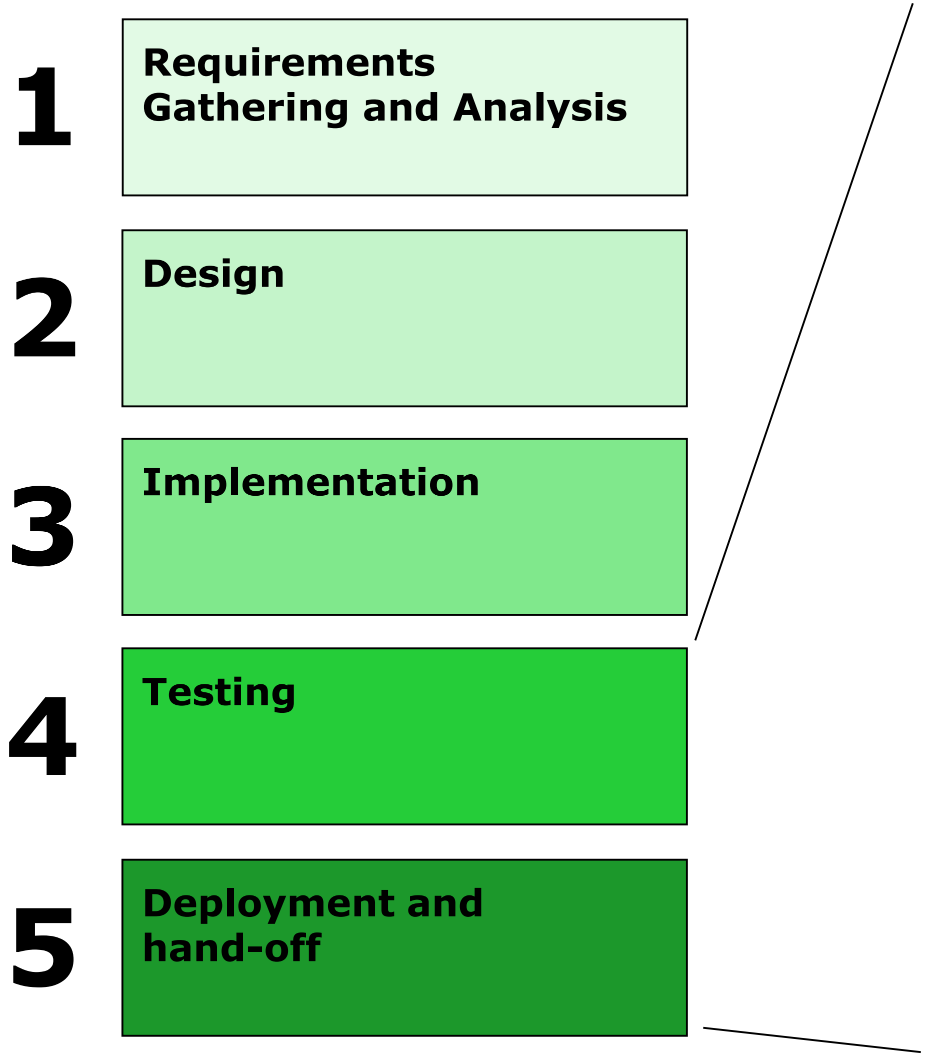
4 **Testing**

5 **Deployment and hand-off**

Implementation

- Create Activities
  - Schema
  - Source and Target
  - Mapping
  - Workflow
  - Events
  - Triggers
  - Notification
- Exception handling
- Design process flow
  - Sequence set of activities
  - Decision nodes
  - Control flows
  - Data flows
- Prepare production environment
  - Size implementation scenario to estimate hardware requirements
  - Requisition hardware and software
  - Deploy hardware and software in production environment

Project steps



1 Requirements Gathering and Analysis

2 Design

3 Implementation

4 Testing

5 Deployment and hand-off

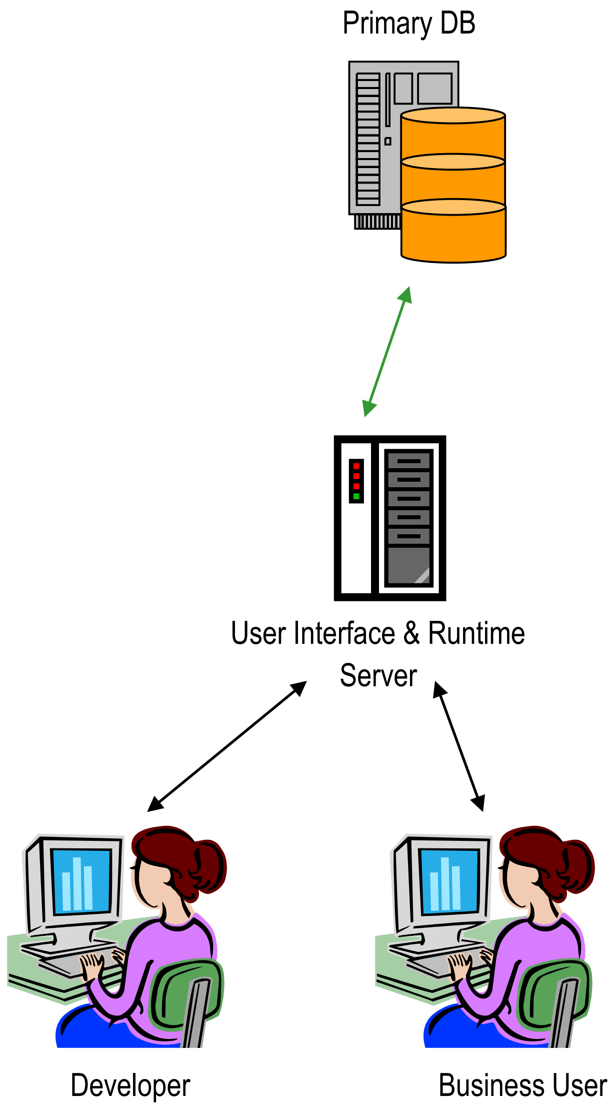
Testing

- Functional testing
  - Test decision nodes with different parameters
  - Test different paths of the flow
  - Test with different datasets
  - Workflow activities testing with file upload and downloads
- Volume testing
  - High data volume
  - High number of simultaneous execution
- User acceptance testing (UAT)
  - Help users design test cases based on their expectations
  - Execute acceptance test cases

Deployment and hand-off

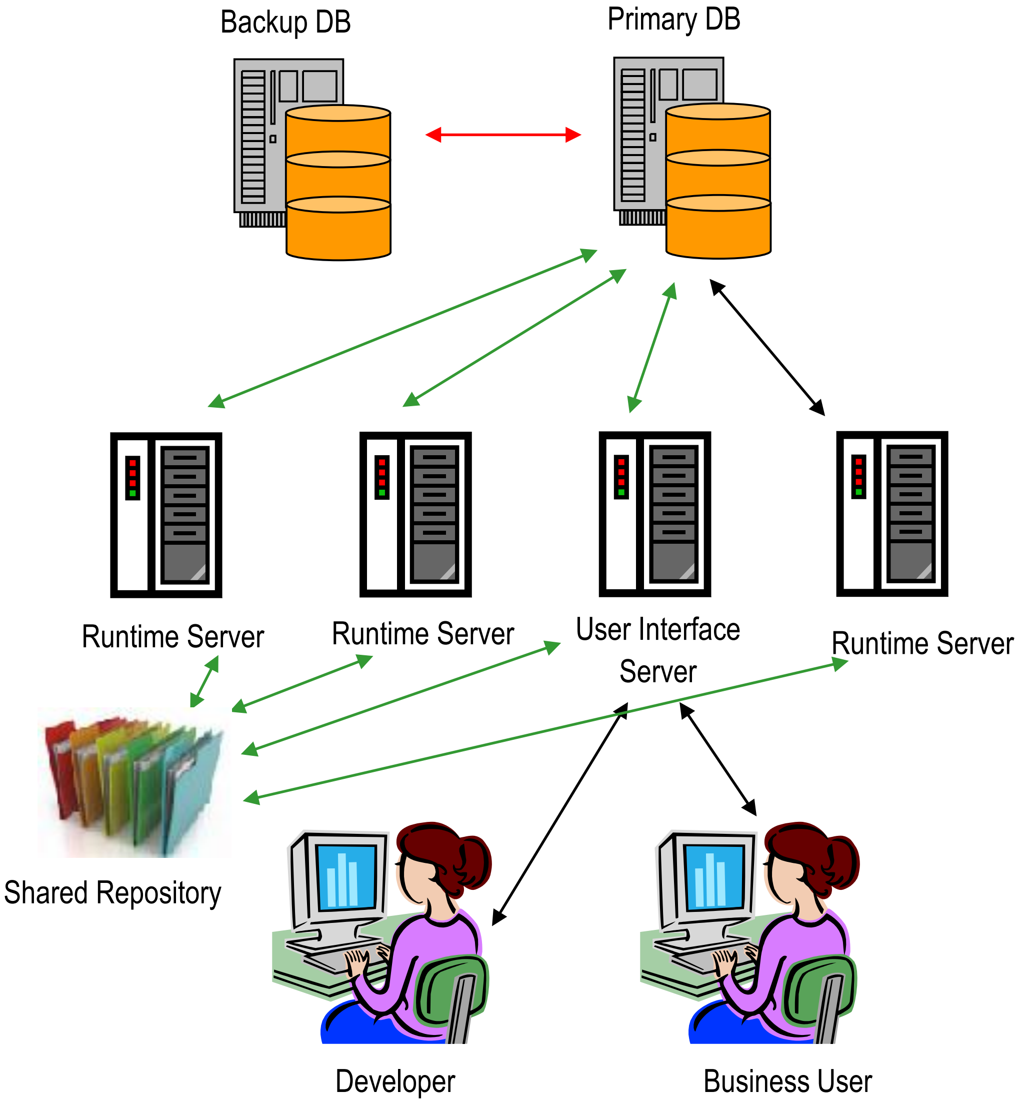
- Deploy at production environment
- Validate deployment with smoke-screen test cases
- Train system admin
- Train business users

Installation Process: Single Server configuration



1. Install Database
   - Cluster DB's for reliability
2. Setup server hardware
   - Multiple servers can be clustered
3. Run Installation Wizard
   1. Installs Java Virtual Machine
   2. Installs Adeptia Suite product binaries
   3. Configure performance parameters
   4. Configure admin parameters
   5. Create Users and Groups

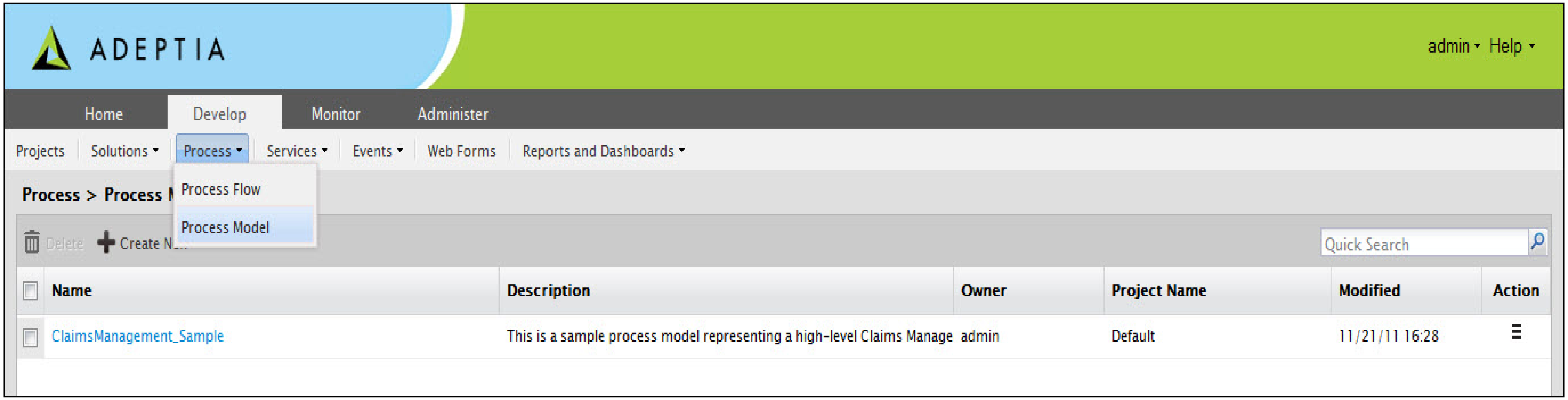
Installation Process: Clustered configuration



1. Install Database
   - Cluster DB's for reliability
2. Setup server hardware
   - Multiple servers can be clustered
3. Configure the shared folder for repository
4. Run Installation Wizard
   1. Installs Java Virtual Machine
   2. Installs Adeptia Suite product binaries
   3. Configure performance parameters
   4. Configure admin parameters
   5. Create Users and Groups

Adeptia Suite Editions

	ENTERPRISE	PREMIER	PROFESSIONAL	EXPRESS
Description	Comprehensive Integration and BPM solution for centrally managing and running all integration interfaces, services and process flows from one place.	Integration-Centric BPM Suite. Both Human2Human and System2System flows. Supports full Process Lifecycle.	Complete Application Integration, SOA Enablement and ESB solution	ETL product focused on data aggregation and transformation. Ideal for BI & Data warehousing needs.
Common Use	All-in-one complete solution development platform	Solution Development Platform for developers to build web-based, data-centric business solutions	Design flexible, custom flows. Any-to-Any Integration, Services Enablement.	App-to-app and B2B integration connections
Process Modeler	Yes	No	No	No
Process Simulation	Yes	No	No	No
Process Engine / Designer	Yes	Yes	Yes	No
Forms	Yes	Limited	No	No
Human Workflow	Yes	Limited	No	No
Reports & Dashboard	Yes	Yes	Limited	No
Services Repository	Yes	Yes	Yes	Yes
Web Services	Yes	Yes	Limited	No
Data Transport	Yes	Yes	Yes	Limited
Events	Yes	Yes	Yes	Limited
Schema & Metadata	Yes	Yes	Yes	Yes
Data Mapping	Yes	Yes	Yes	Yes
Exception/Error Handling	Yes	Yes	Yes	No
Upgrade path	Single install, Product upgraded based on license key only			

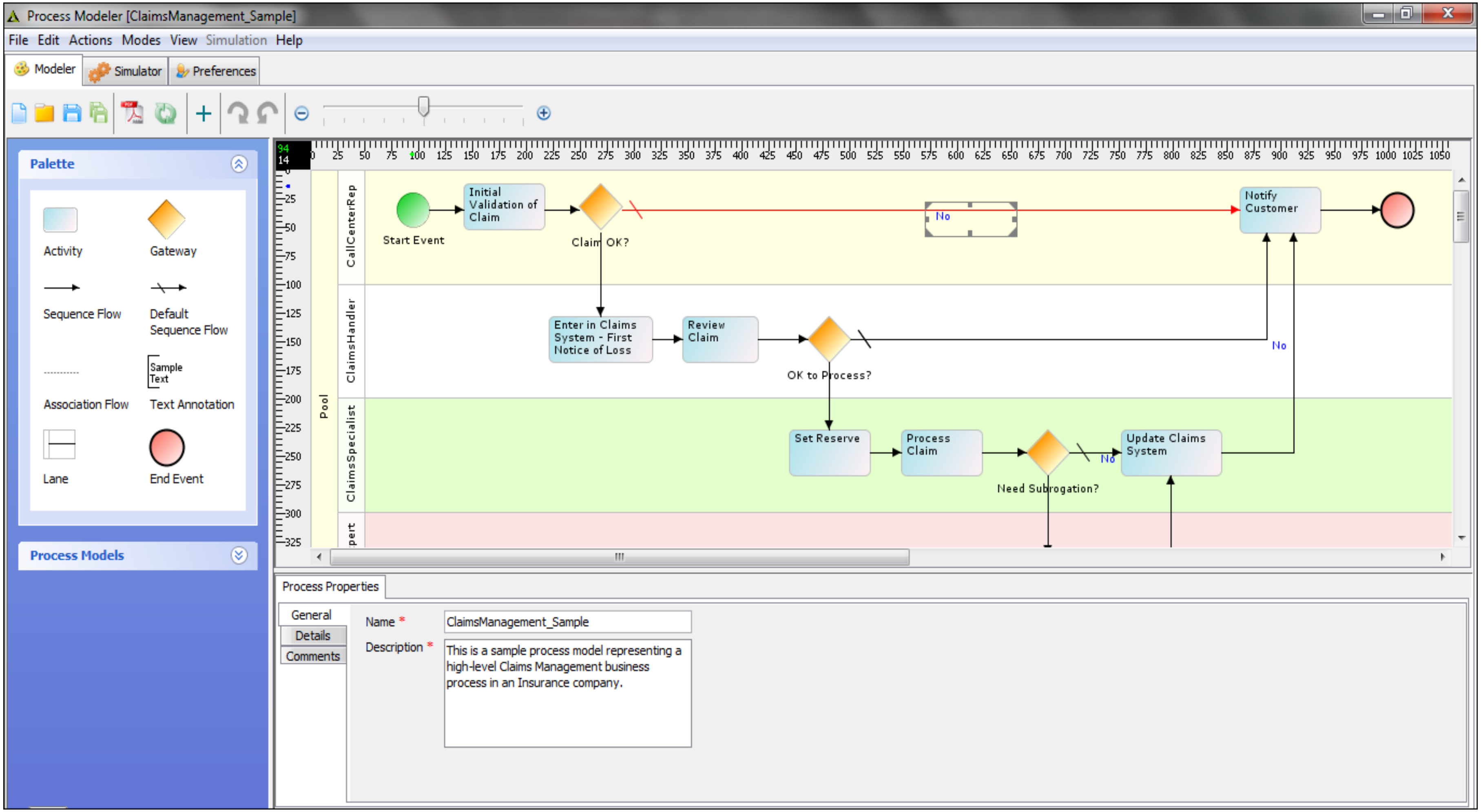


Process > Process M

□	Name	Description	Owner	Project Name	Modified	Action
□	ClaimsManagement_Sample	This is a sample process model representing a high-level Claims Manage	admin	Default	11/21/11 16:28	☰

Process Model

Exercise: Model a Process



Name * ClaimsManagement_Sample

Description * This is a sample process model representing a high-level Claims Management business process in an Insurance company.

Refer to Notes below

Simulation

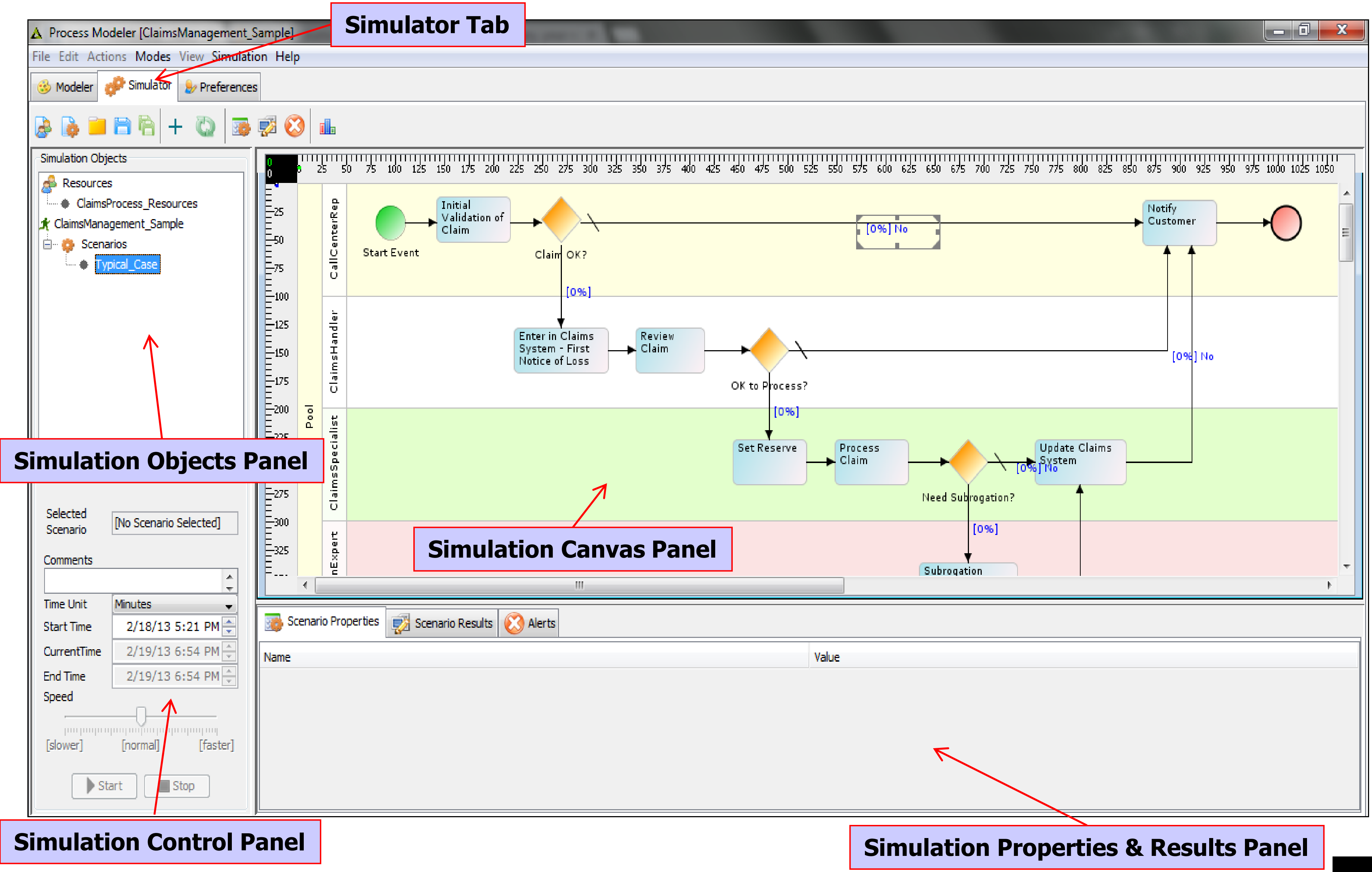
Simulation - Introduction

- Some of the important issues to an organization are increasing efficiency, reducing costs, and improving process effectiveness and innovation, as well as standardizing and harmonizing business processes. These issues can be resolved by modeling processes using a business process management tool. However, implementing these processes may lead to low process performance instead of yielding expected gains.

- The objective of creating an efficient Process can be achieved by using Business Process Simulation, which is an ideal tool to achieve optimization by enhancing performance and reducing cost.

- Simulating Business Process Flow involves following steps:
  1. Design a Process Flow model using the Process Designer
  2. Resources are identified and assigned to the Process Flow Activities
  3. Simulation Scenarios are defined and then executed
  4. When the simulation is completed, the statistical results are analyzed

Simulation – Process Simulator Interface



Simulator Tab

Simulation Objects Panel

Simulation Canvas Panel

Simulation Control Panel

Simulation Properties & Results Panel

Selected Scenario: [No Scenario Selected]

Time Unit: Minutes

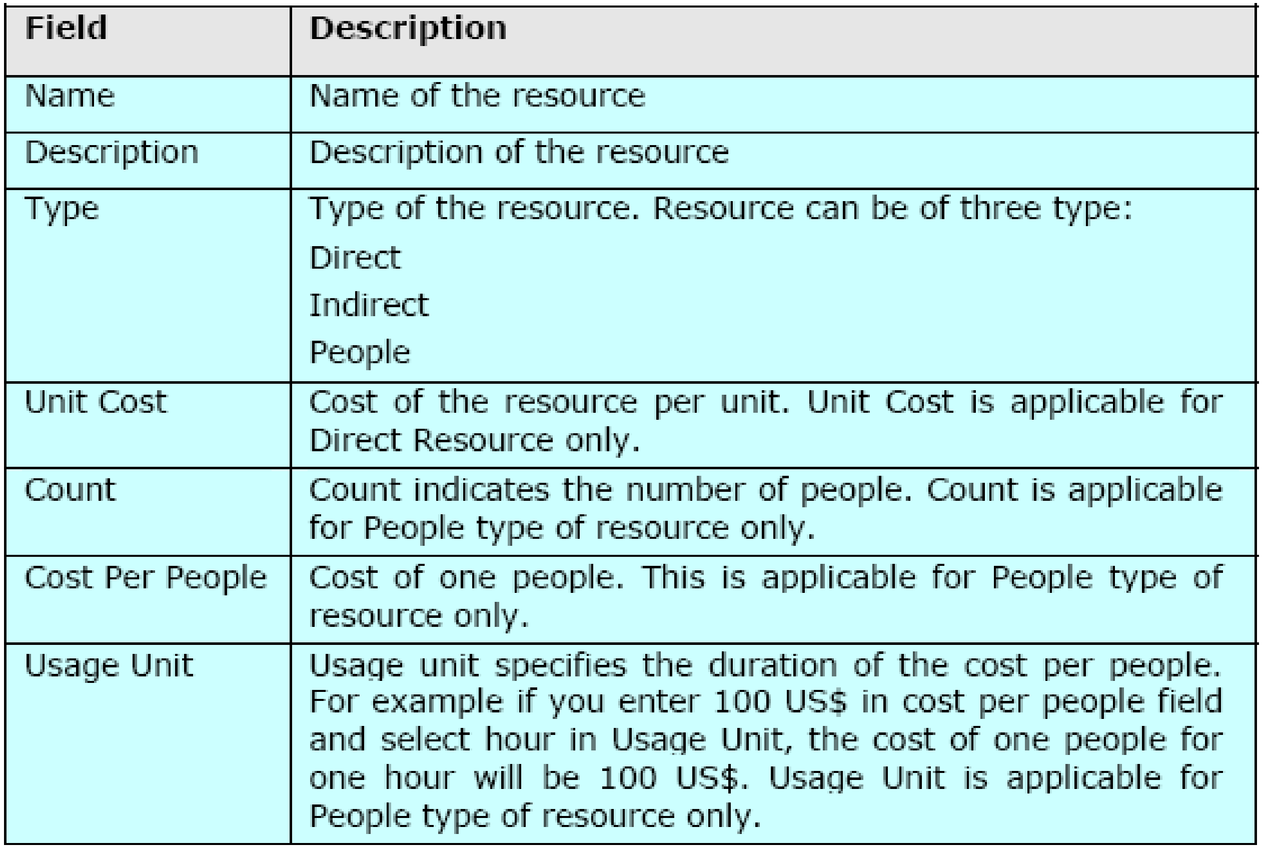
Start Time: 2/18/13 5:21 PM

CurrentTime: 2/19/13 6:54 PM

End Time: 2/19/13 6:54 PM

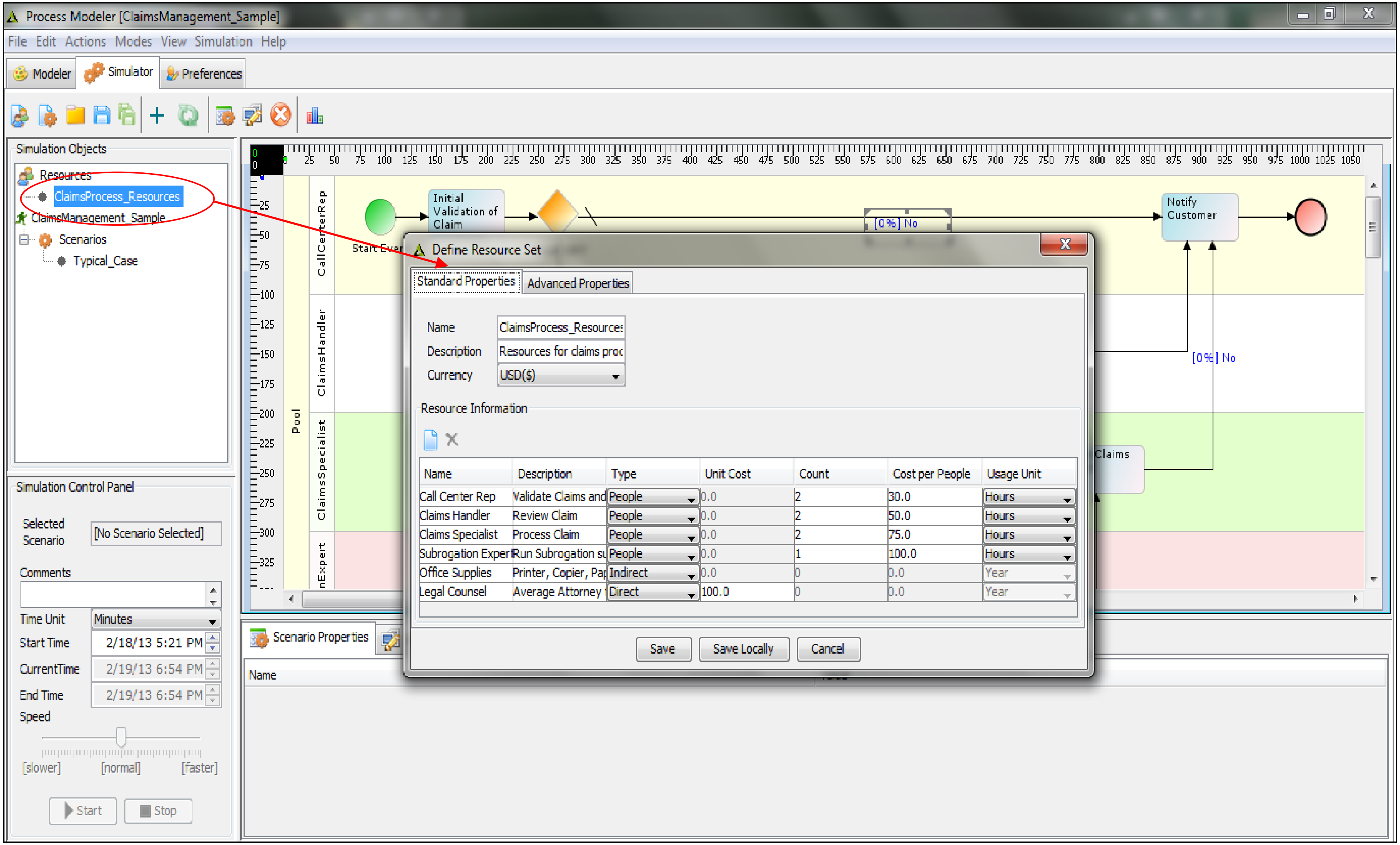
Simulation – Step 2 : Define a Resource Set

Resource Set attributes:



Field	Description
Name	Name of the resource
Description	Description of the resource
Type	Type of the resource. Resource can be of three type: Direct Indirect People
Unit Cost	Cost of the resource per unit. Unit Cost is applicable for Direct Resource only.
Count	Count indicates the number of people. Count is applicable for People type of resource only.
Cost Per People	Cost of one people. This is applicable for People type of resource only.
Usage Unit	Usage unit specifies the duration of the cost per people. For example if you enter 100 US$ in cost per people field and select hour in Usage Unit, the cost of one people for one hour will be 100 US$. Usage Unit is applicable for People type of resource only.

Simulation – Step 2 : Define a Resource Set



Selected Scenario: [No Scenario Selected]

Comments:

Time Unit: Minutes

Start Time: 2/18/13 5:21 PM

CurrentTime: 2/19/13 6:54 PM

End Time: 2/19/13 6:54 PM

Speed

Define Resource Set – Standard Properties

Name: ClaimsProcess_Resources

Description: Resources for claims proc

Currency: USD($)

Resource Information

Name	Description	Type	Unit Cost	Count	Cost per People	Usage Unit
Call Center Rep	Validate Claims and	People	0.0	2	30.0	Hours
Claims Handler	Review Claim	People	0.0	2	50.0	Hours
Claims Specialist	Process Claim	People	0.0	2	75.0	Hours
Subrogation Exper	Run Subrogation su	People	0.0	1	100.0	Hours
Office Supplies	Printer, Copier, Pap	Indirect	0.0	0	0.0	Year
Legal Counsel	Average Attorney	Direct	100.0	0	0.0	Year

Refer to Notes below

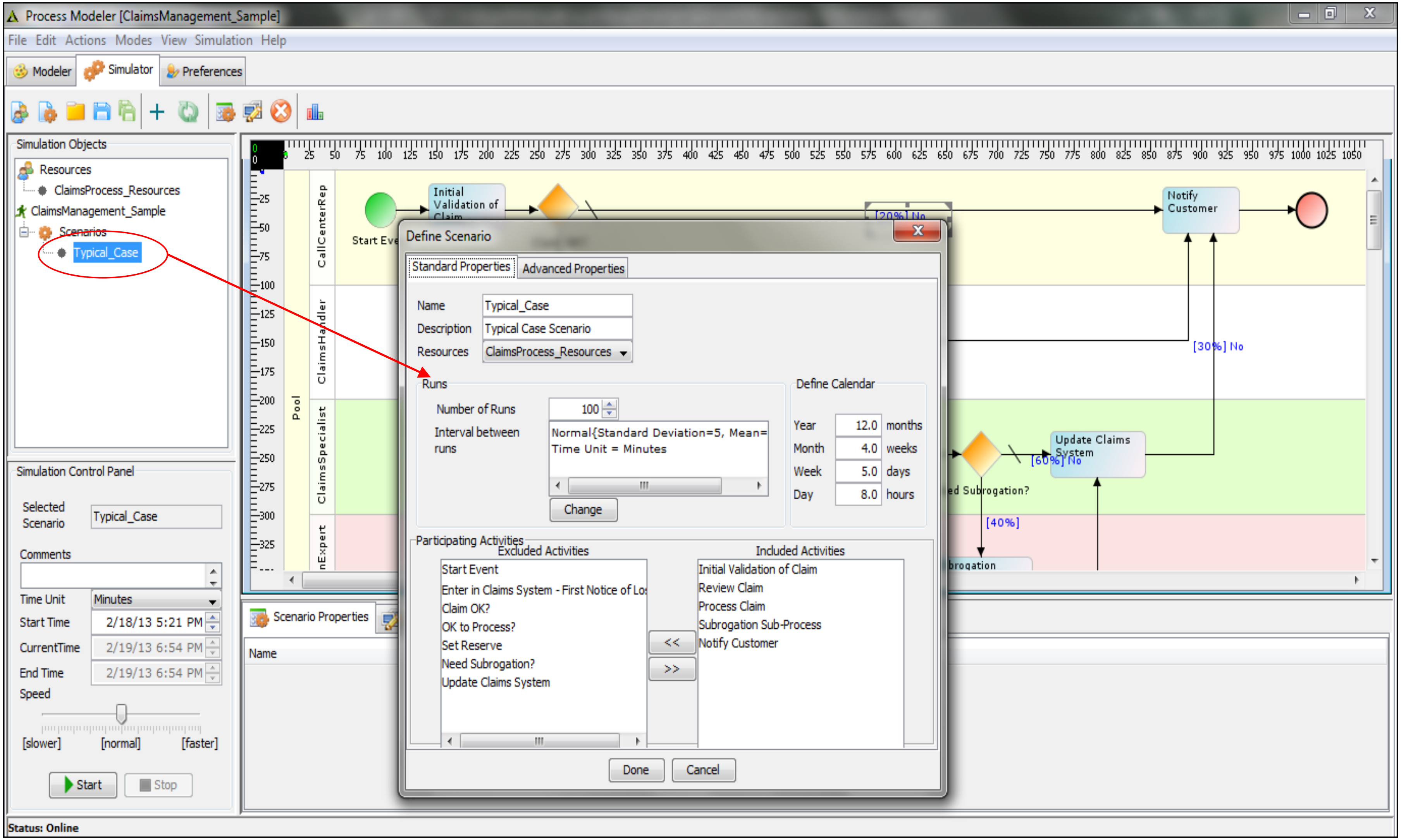
Simulation – Step 3 : Define a Scenario

- A Scenario contains a complete set of data used to simulate a business process.

- It includes resources used by the Activities of the Process Flow, time used by each Activity, randomness, and the number of times the Process Flow will be executed to get the simulation result.

- A scenario can be used for only one process flow. Multiple scenarios can be used for a single process flow.

Simulation – Step 3 : Define a Scenario



Define Scenario

Standard Properties | Advanced Properties

Name: Typical_Case

Description: Typical Case Scenario

Resources: ClaimsProcess_Resources

Runs

Number of Runs: 100

Interval between runs: Normal{Standard Deviation=5, Mean= Time Unit = Minutes

Change

Define Calendar

Year	12.0	months
Month	4.0	weeks
Week	5.0	days
Day	8.0	hours

Participating Activities

Excluded Activities	Included Activities
Start Event	Initial Validation of Claim
Enter in Claims System - First Notice of Lo:	Review Claim
Claim OK?	Process Claim
OK to Process?	Subrogation Sub-Process
Set Reserve	Notify Customer
Need Subrogation?	
Update Claims System	

<< >>

Done Cancel

Simulation Control Panel

Selected Scenario: Typical_Case

Comments:

Time Unit: Minutes

Start Time: 2/18/13 5:21 PM

CurrentTime: 2/19/13 6:54 PM

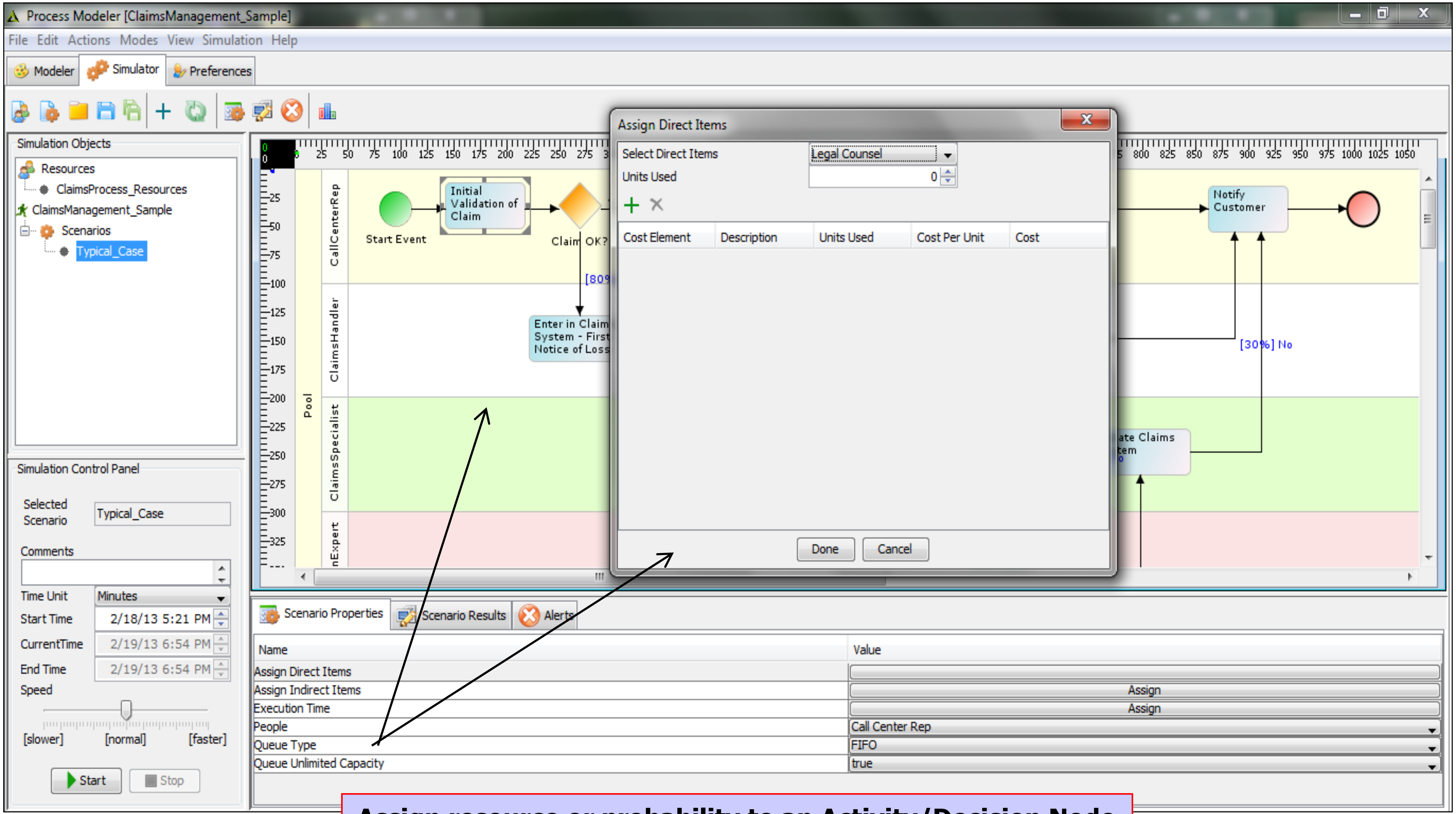
End Time: 2/19/13 6:54 PM

Speed: [slower] [normal] [faster]

Start Stop

Refer to Notes below

Simulation – Step 4 : Assigning Resource and/or Probability to Process Flow



Assign Direct Items

Select Direct Items: Legal Counsel

Units Used: 0

Cost Element	Description	Units Used	Cost Per Unit	Cost

Selected Scenario: Typical_Case

Comments:

Time Unit: Minutes

Start Time: 2/18/13 5:21 PM

CurrentTime: 2/19/13 6:54 PM

End Time: 2/19/13 6:54 PM

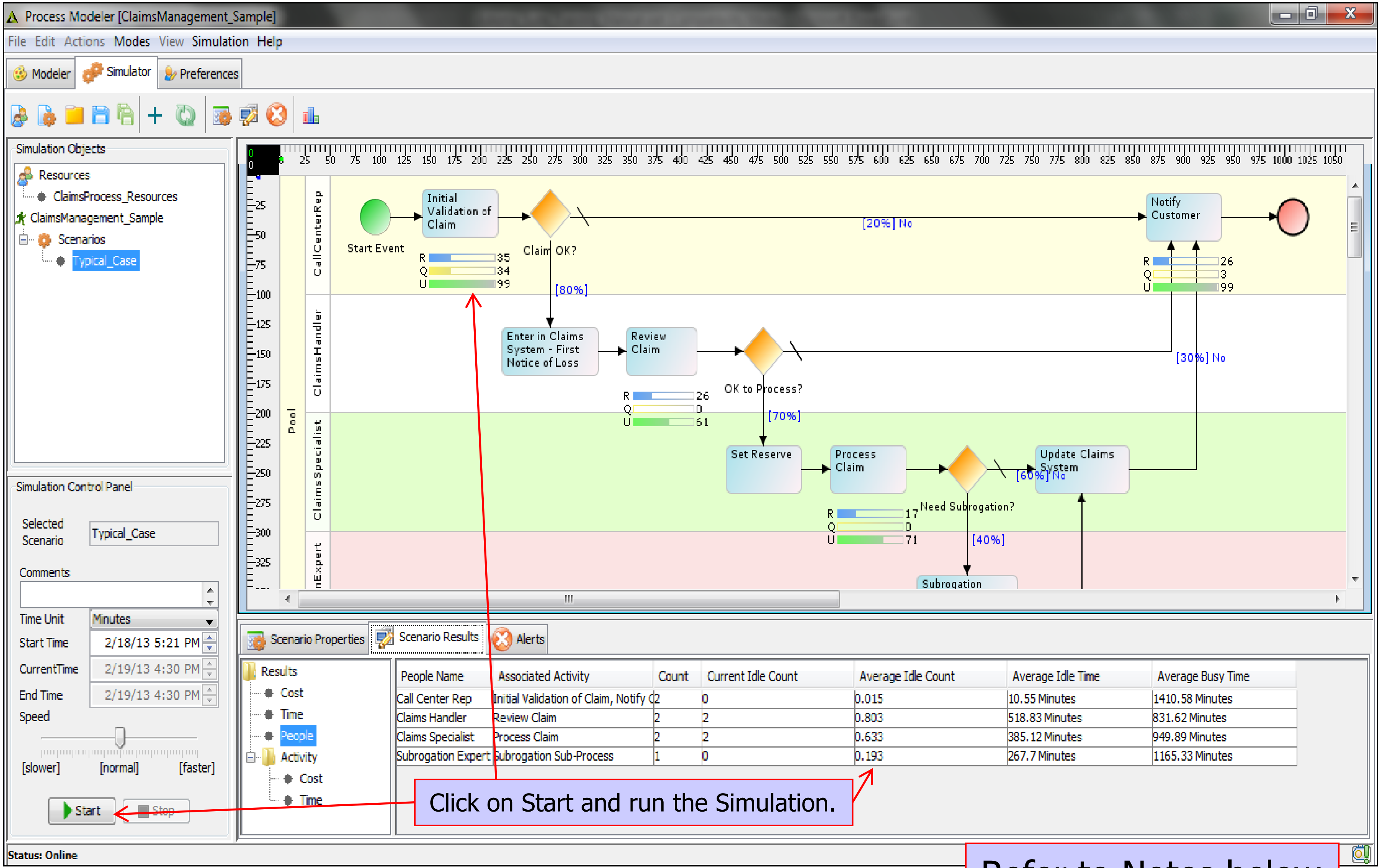
Speed

Name	Value
Assign Direct Items	
Assign Indirect Items	Assign
Execution Time	Assign
People	Call Center Rep
Queue Type	FIFO
Queue Unlimited Capacity	true

Assign resource or probability to an Activity/Decision Node

Refer to Notes below

Simulation – Step 5 : Executing Simulation



Selected Scenario: Typical_Case

Time Unit: Minutes

Start Time: 2/18/13 5:21 PM

CurrentTime: 2/19/13 4:30 PM

End Time: 2/19/13 4:30 PM

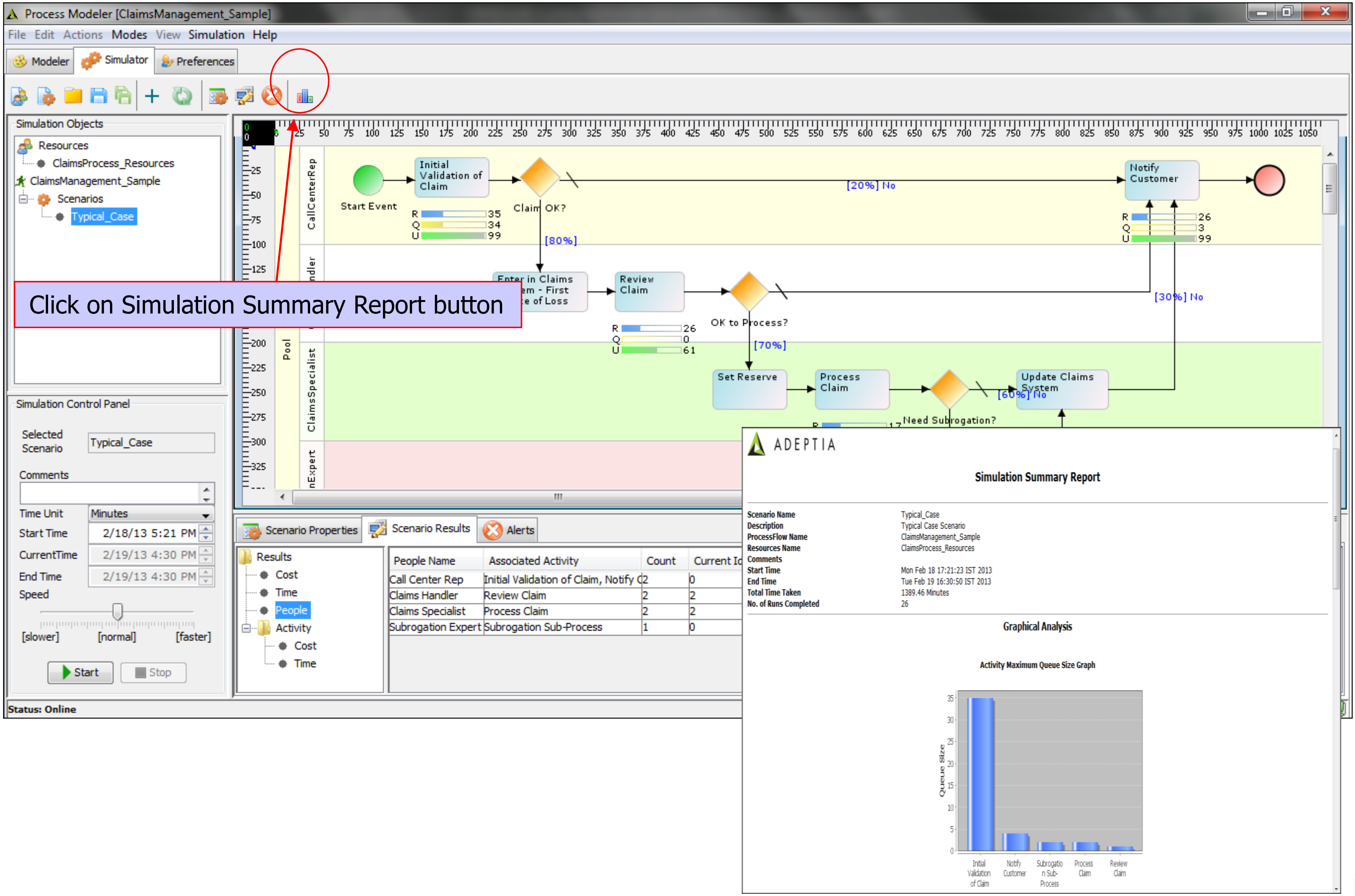
People Name	Associated Activity	Count	Current Idle Count	Average Idle Count	Average Idle Time	Average Busy Time
Call Center Rep	Initial Validation of Claim, Notify C	2	0	0.015	10.55 Minutes	1410.58 Minutes
Claims Handler	Review Claim	2	2	0.803	518.83 Minutes	831.62 Minutes
Claims Specialist	Process Claim	2	2	0.633	385.12 Minutes	949.89 Minutes
Subrogation Expert	Subrogation Sub-Process	1	0	0.193	267.7 Minutes	1165.33 Minutes

Status: Online

Click on Start and run the Simulation.

Refer to Notes below

Simulation – Step 6 : Analyze Simulation Summary Report



Selected Scenario: Typical_Case

Comments:

Time Unit: Minutes

Start Time: 2/18/13 5:21 PM

CurrentTime: 2/19/13 4:30 PM

End Time: 2/19/13 4:30 PM

Speed: [slower] [normal] [faster]

People Name	Associated Activity	Count	Current Id
Call Center Rep	Initial Validation of Claim, Notify C	2	0
Claims Handler	Review Claim	2	2
Claims Specialist	Process Claim	2	2
Subrogation Expert	Subrogation Sub-Process	1	0

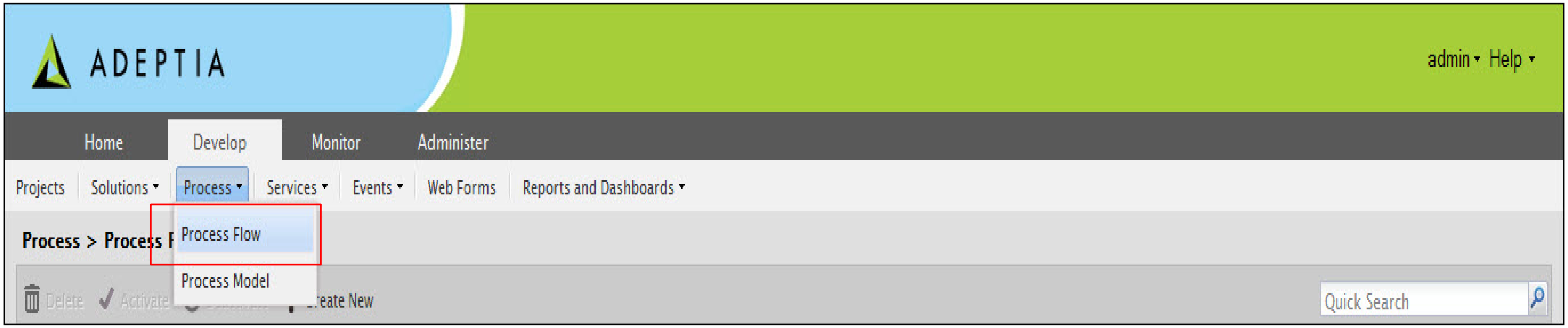
Status: Online

Simulation Summary Report

Scenario Name	Typical_Case
Description	Typical Case Scenario
ProcessFlow Name	ClaimsManagement_Sample
Resources Name	ClaimsProcess_Resources
Comments	
Start Time	Mon Feb 18 17:21:23 IST 2013
End Time	Tue Feb 19 16:30:50 IST 2013
Total Time Taken	1389.46 Minutes
No. of Runs Completed	26

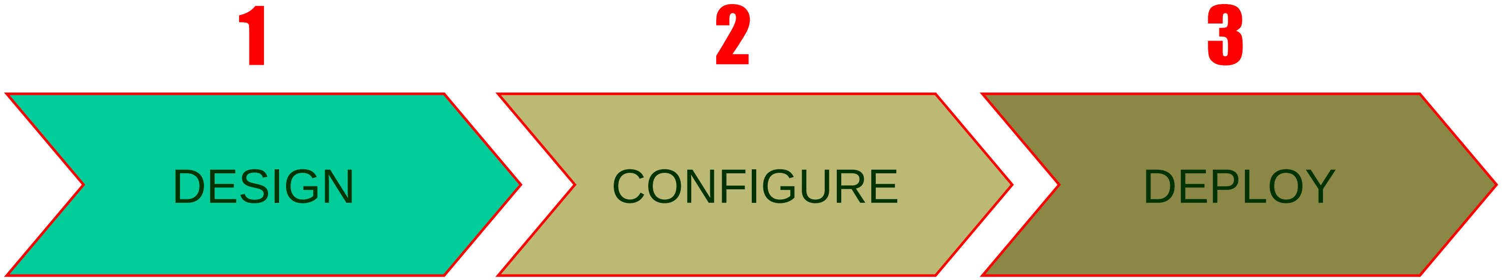
Graphical Analysis

Click on Simulation Summary Report button



Process Designer

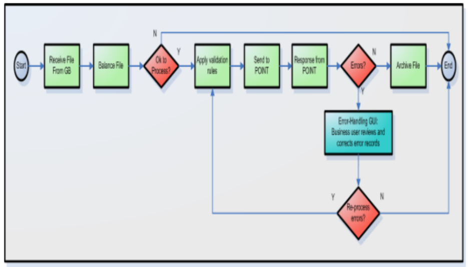
Using Adeptia: 3 Simple Steps



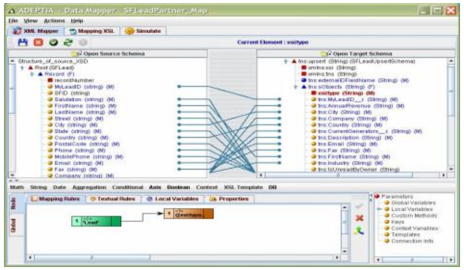
1 DESIGN

2 CONFIGURE

3 DEPLOY



- Create / modify process flows
- Define business and data rules
- Define exceptions & error handling



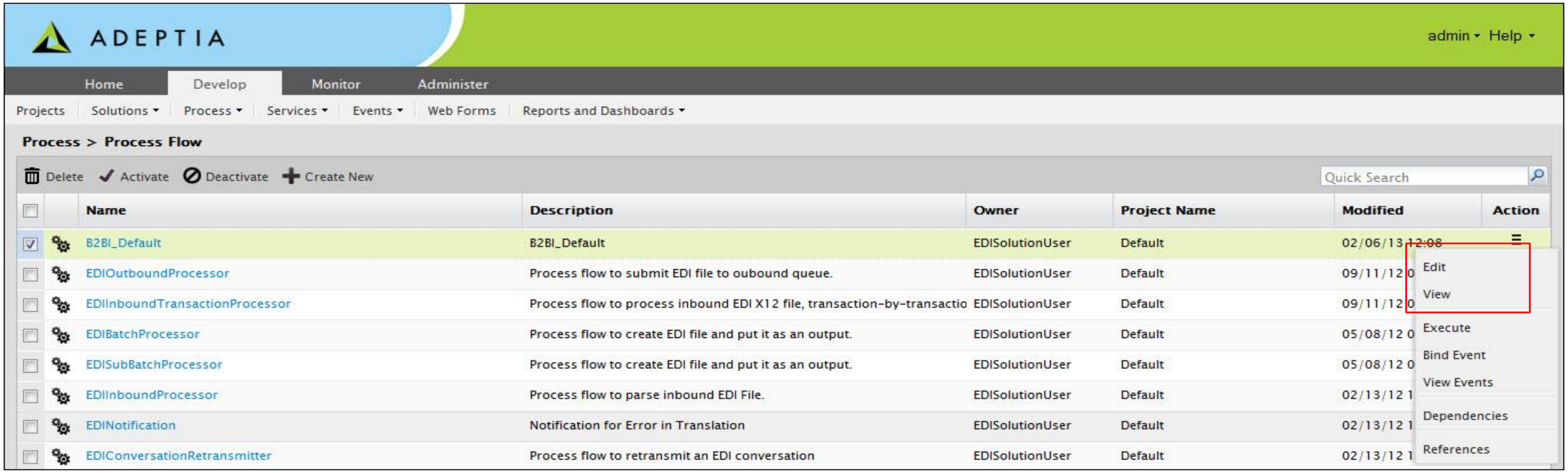
- Configure data access parameters
- Configure back-end app schemas
- Configure data mapping rules

- Flows & Services are in server repository
- Create a real time or batch-mode trigger
- Attach trigger to process flow & activate

Process Designer

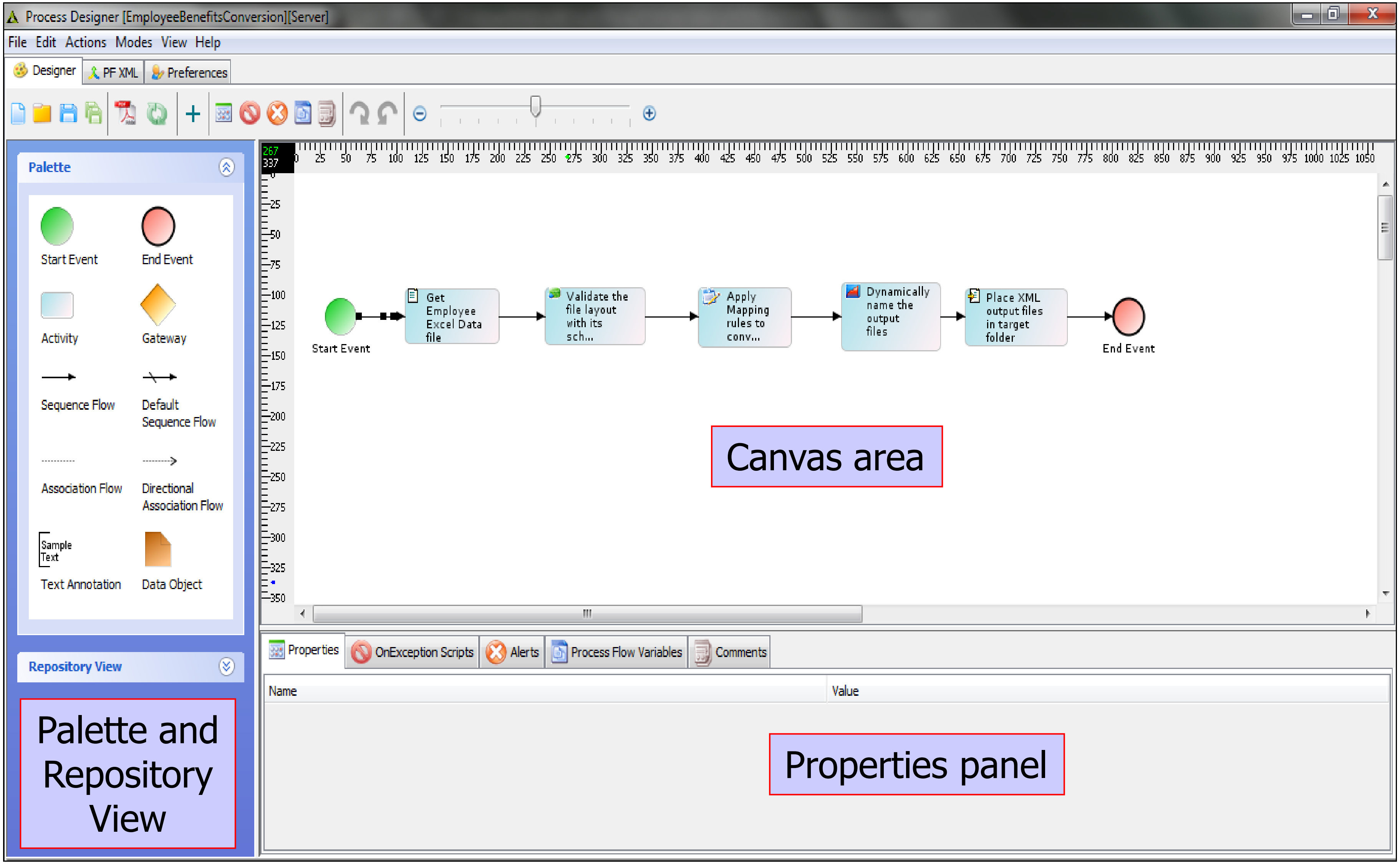
- Adeptia Suite offers a graphical process design interface.
- Simple to use, drag and drop interface for business managers
- Allows both “top-down” and “bottom-up” approach for building processes
- Includes a powerful, full-featured engine for process execution
- Process engine logs all runtime information

Go to Develop> Process >Process Flow
Manage page will show a list of all processes that are available for edit or execution.
List will show process name, description, owner and last modified.
Select a process by clicking on the check box and click on Edit from Action menu.
Edit screen will have a Process Designer button which will open up the process flow.
User can click on Create New to create a new process flow.



□		Name	Description	Owner	Project Name	Modified	Action
☑		B2BI_Default	B2BI_Default	EDISolutionUser	Default	02/06/13 12:08	≡
□		EDIOutboundProcessor	Process flow to submit EDI file to oubound queue.	EDISolutionUser	Default	09/11/12 0	
□		EDIInboundTransactionProcessor	Process flow to process inbound EDI X12 file, transaction-by-transactio	EDISolutionUser	Default	09/11/12 0	
□		EDIBatchProcessor	Process flow to create EDI file and put it as an output.	EDISolutionUser	Default	05/08/12 0	
□		EDISubBatchProcessor	Process flow to create EDI file and put it as an output.	EDISolutionUser	Default	05/08/12 0	
□		EDIInboundProcessor	Process flow to parse inbound EDI File.	EDISolutionUser	Default	02/13/12 1	
□		EDINotification	Notification for Error in Translation	EDISolutionUser	Default	02/13/12 1	
□		EDIConversationRetransmitter	Process flow to retransmit an EDI conversation	EDISolutionUser	Default	02/13/12 1	

Process Designer Overview

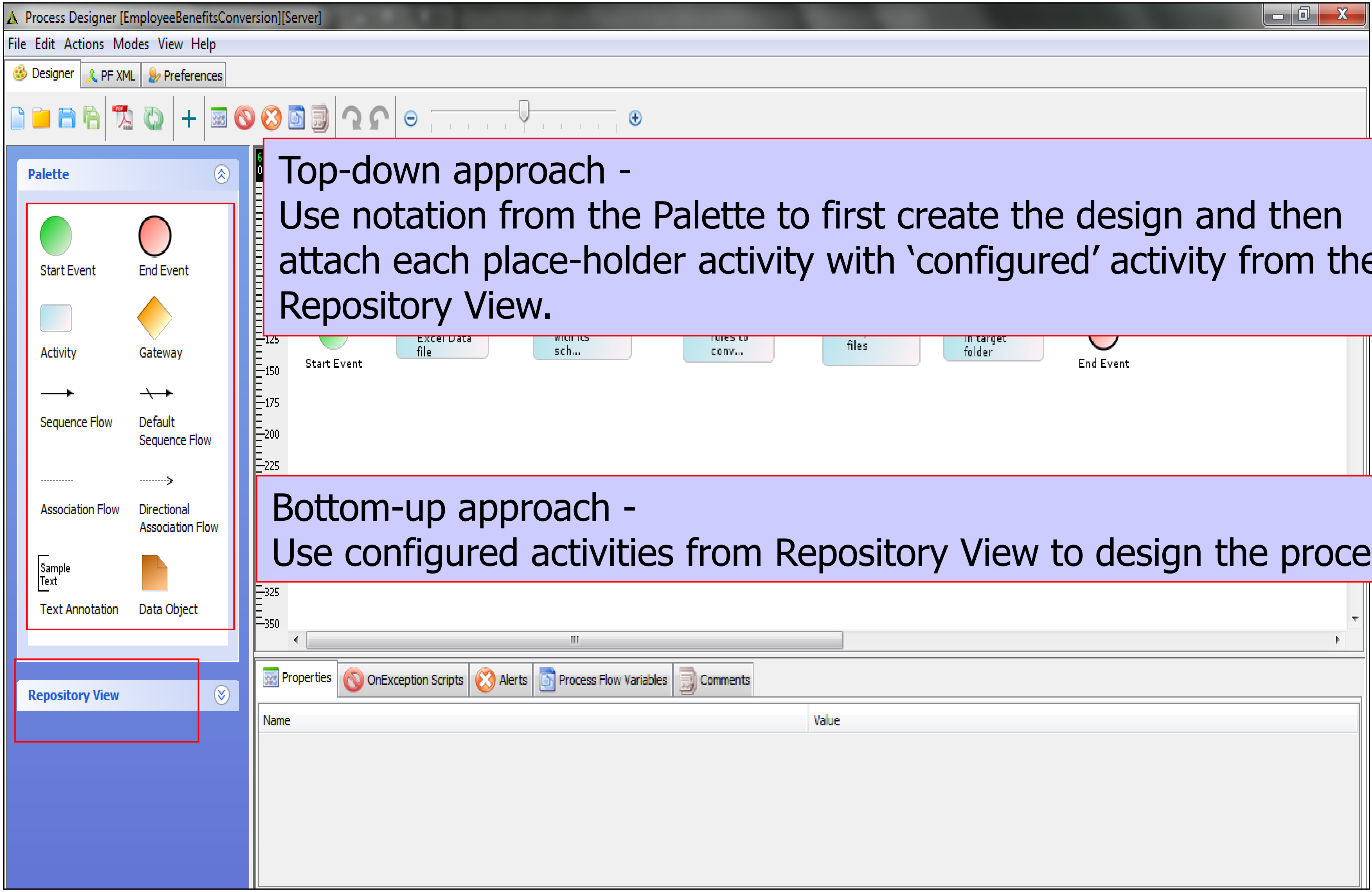

Process Designer [EmployeeBenefitsConversion][Server]
File Edit Actions Modes View Help
Designer
PF XML
Preferences
Palette
Start Event
End Event
Activity
Gateway
Sequence Flow
Default Sequence Flow
Association Flow
Directional Association Flow
Sample Text
Text Annotation
Data Object
Repository View
Start Event
Get Employee Excel Data file
Validate the file layout with its sch...
Apply Mapping rules to conv...
Dynamically name the output files
Place XML output files in target folder
End Event
Properties
OnException Scripts
Alerts
Process Flow Variables
Comments
Name
Value

Canvas area

Palette and Repository View

Properties panel

Process Designer Overview

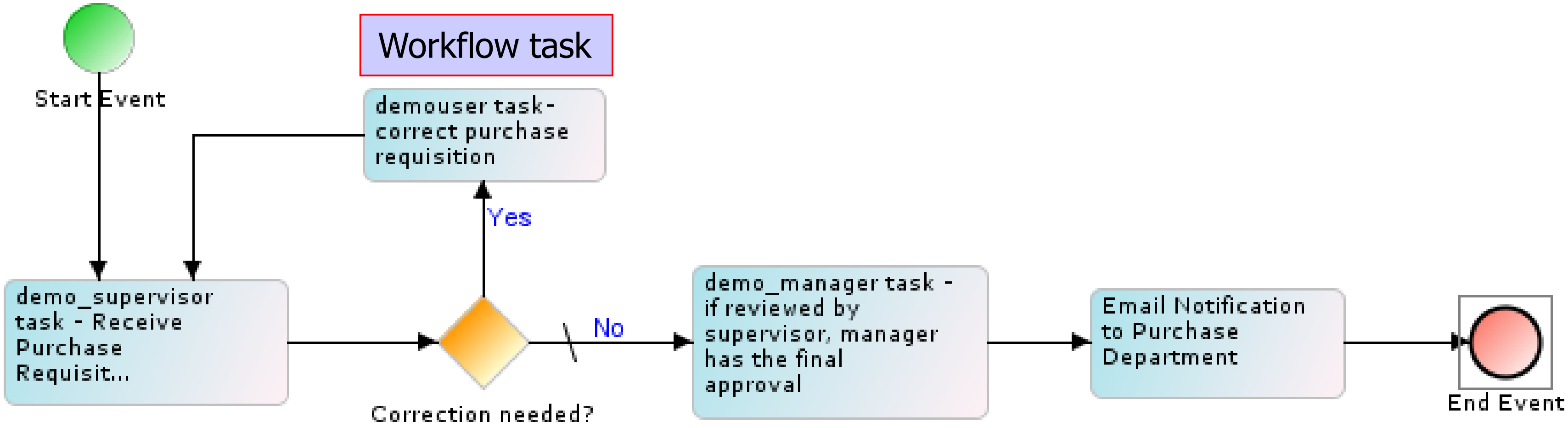


Top-down approach -
Use notation from the Palette to first create the design and then attach each place-holder activity with ‘configured’ activity from the Repository View.

Bottom-up approach -
Use configured activities from Repository View to design the process.

BPM Exercise: Create Workflow

Step 1: Design Purchase Requisition Flow


Start Event
demouser task - correct purchase requisition
Yes
demo_supervisor task - Receive Purchase Requisit...
Correction needed?
No
demo_manager task - if reviewed by supervisor, manager has the final approval
Email Notification to Purchase Department
End Event

Workflow task

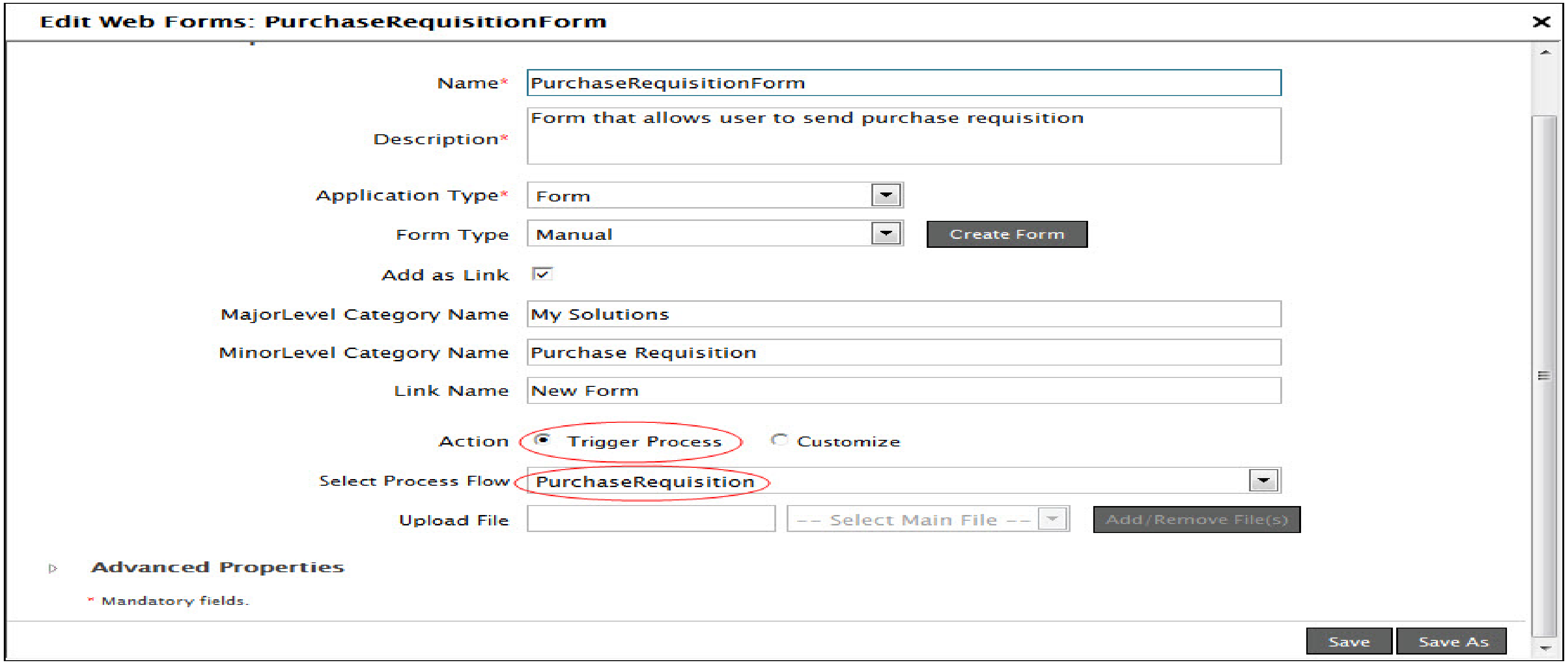
Workflow task

Workflow task

Email notification

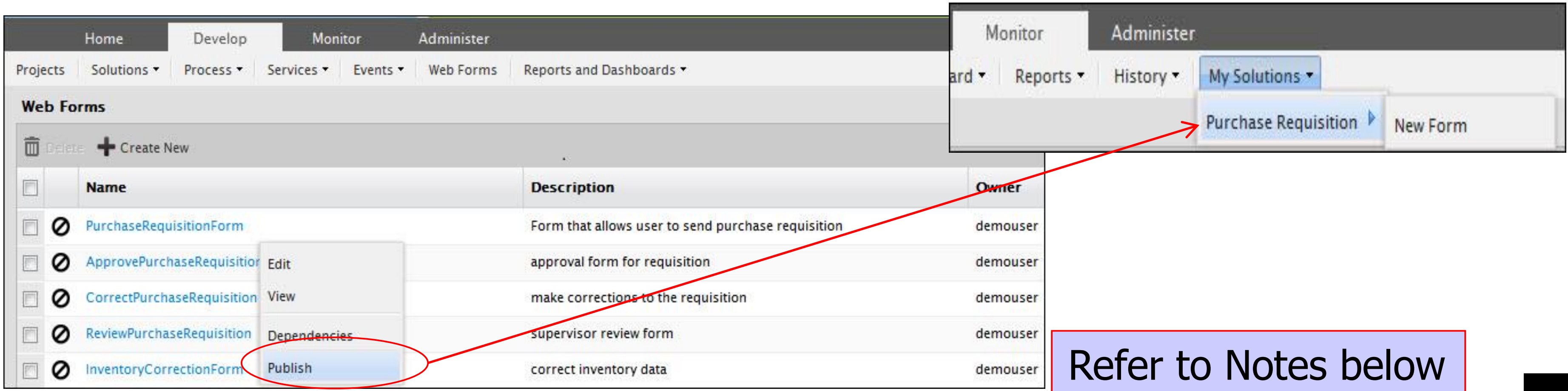
Refer to Notes below

Step 2: Create Purchase Requisition Web Form



Edit Web Forms: PurchaseRequisitionForm

Name*	PurchaseRequisitionForm
Description*	Form that allows user to send purchase requisition
Application Type*	Form
Form Type	Manual
Add as Link	☑
MajorLevel Category Name	My Solutions
MinorLevel Category Name	Purchase Requisition
Link Name	New Form
Action	● Trigger Process ○ Customize
Select Process Flow	PurchaseRequisition
Upload File	-- Select Main File --

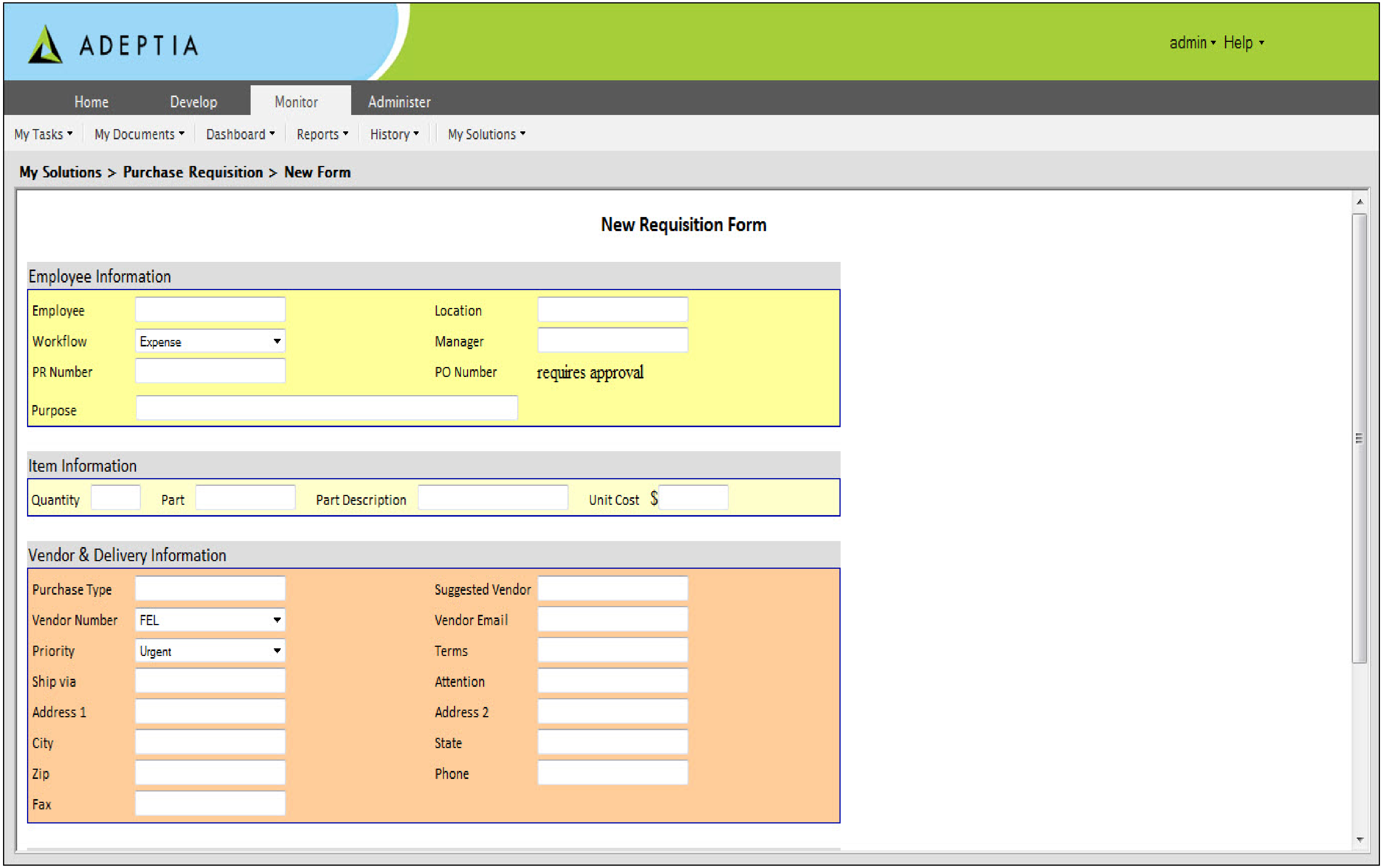


Web Forms

□		Name	Description	Owner
□	⊘	PurchaseRequisitionForm	Form that allows user to send purchase requisition	demouser
□	⊘	ApprovePurchaseRequisitio	approval form for requisition	demouser
□	⊘	CorrectPurchaseRequisition	make corrections to the requisition	demouser
□	⊘	ReviewPurchaseRequisition	supervisor review form	demouser
□	⊘	InventoryCorrectionForm	correct inventory data	demouser

Refer to Notes below

Step 3: Activated Purchase Requisition Web Form



My Solutions > Purchase Requisition > New Form

New Requisition Form

Employee Information

Employee		Location	
Workflow	Expense	Manager	
PR Number		PO Number	requires approval
Purpose			

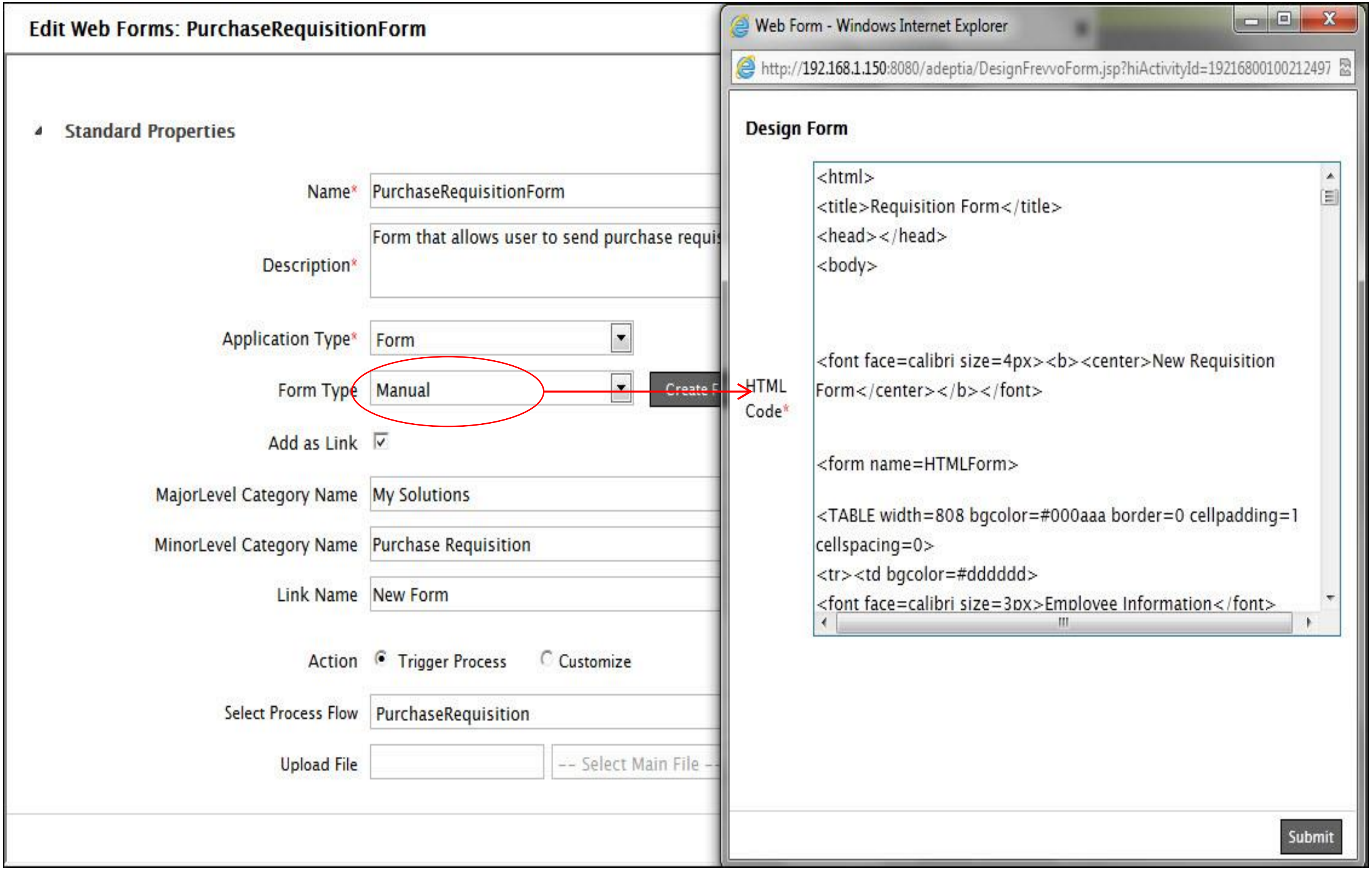
Item Information

Quantity		Part		Part Description		Unit Cost	$

Vendor & Delivery Information

Purchase Type		Suggested Vendor	
Vendor Number	FEL	Vendor Email	
Priority	Urgent	Terms	
Ship via		Attention	
Address 1		Address 2	
City		State	
Zip		Phone	
Fax			

Step 4: Create Review PR Web Form



Edit Web Forms: PurchaseRequisitionForm

Standard Properties

Name*	PurchaseRequisitionForm
Description*	Form that allows user to send purchase requis
Application Type*	Form
Form Type	Manual
Add as Link	☑
MajorLevel Category Name	My Solutions
MinorLevel Category Name	Purchase Requisition
Link Name	New Form
Action	● Trigger Process ○ Customize
Select Process Flow	PurchaseRequisition
Upload File	-- Select Main File --

http://192.168.1.150:8080/adeptia/DesignFrevvoForm.jsp?hiActivityId=19216800100212497

Design Form

HTML Code*

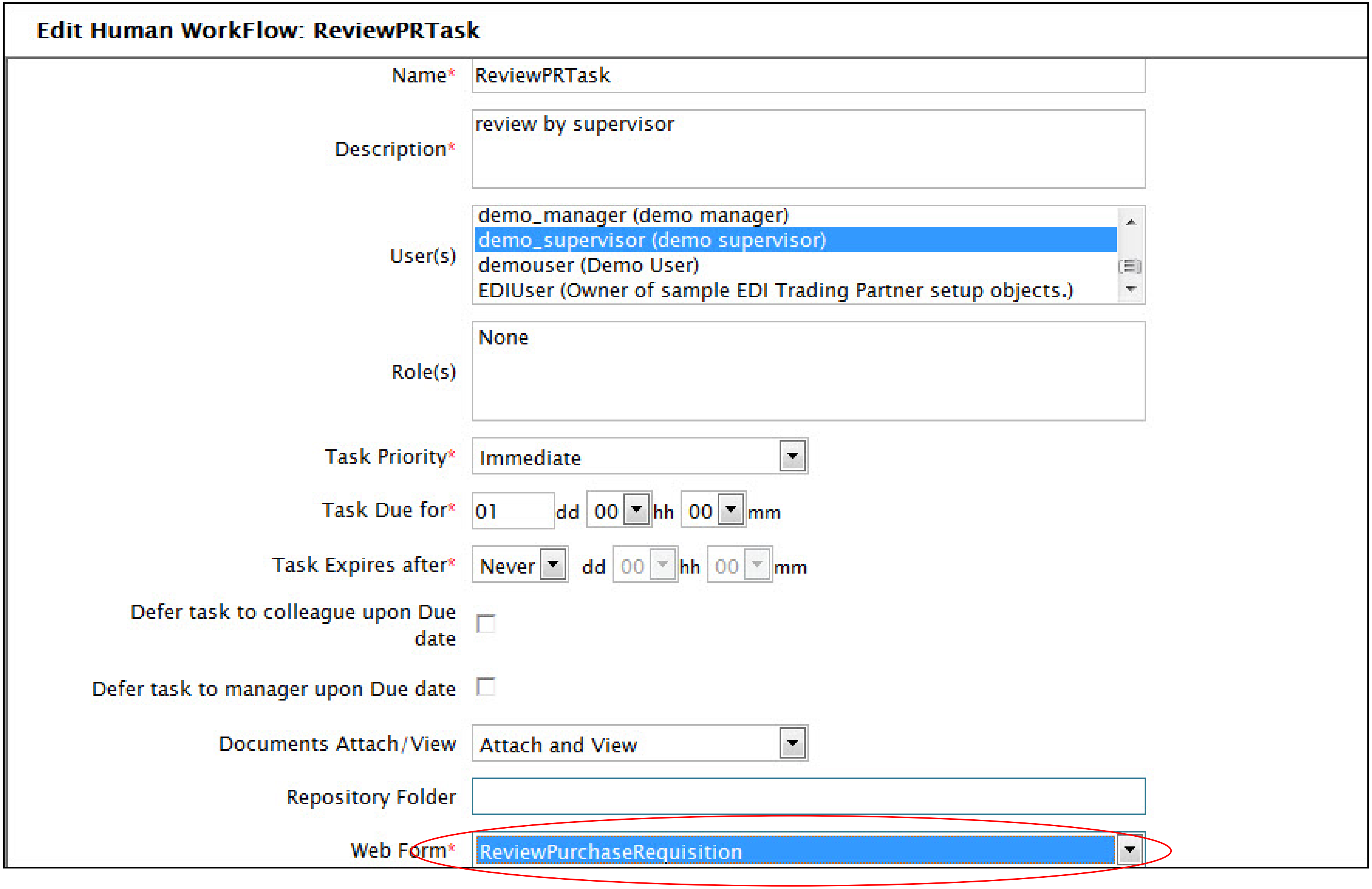
```
<html>
<title>Requisition Form</title>
<head></head>
<body>


<font face=calibri size=4px><b><center>New Requisition
Form</center></b></font>


<form name=HTMLForm>

<TABLE width=808 bgcolor=#000aaa border=0 cellpadding=1
cellspacing=0>
<tr><td bgcolor=#dddddd>
<font face=calibri size=3px>Employee Information</font>
```

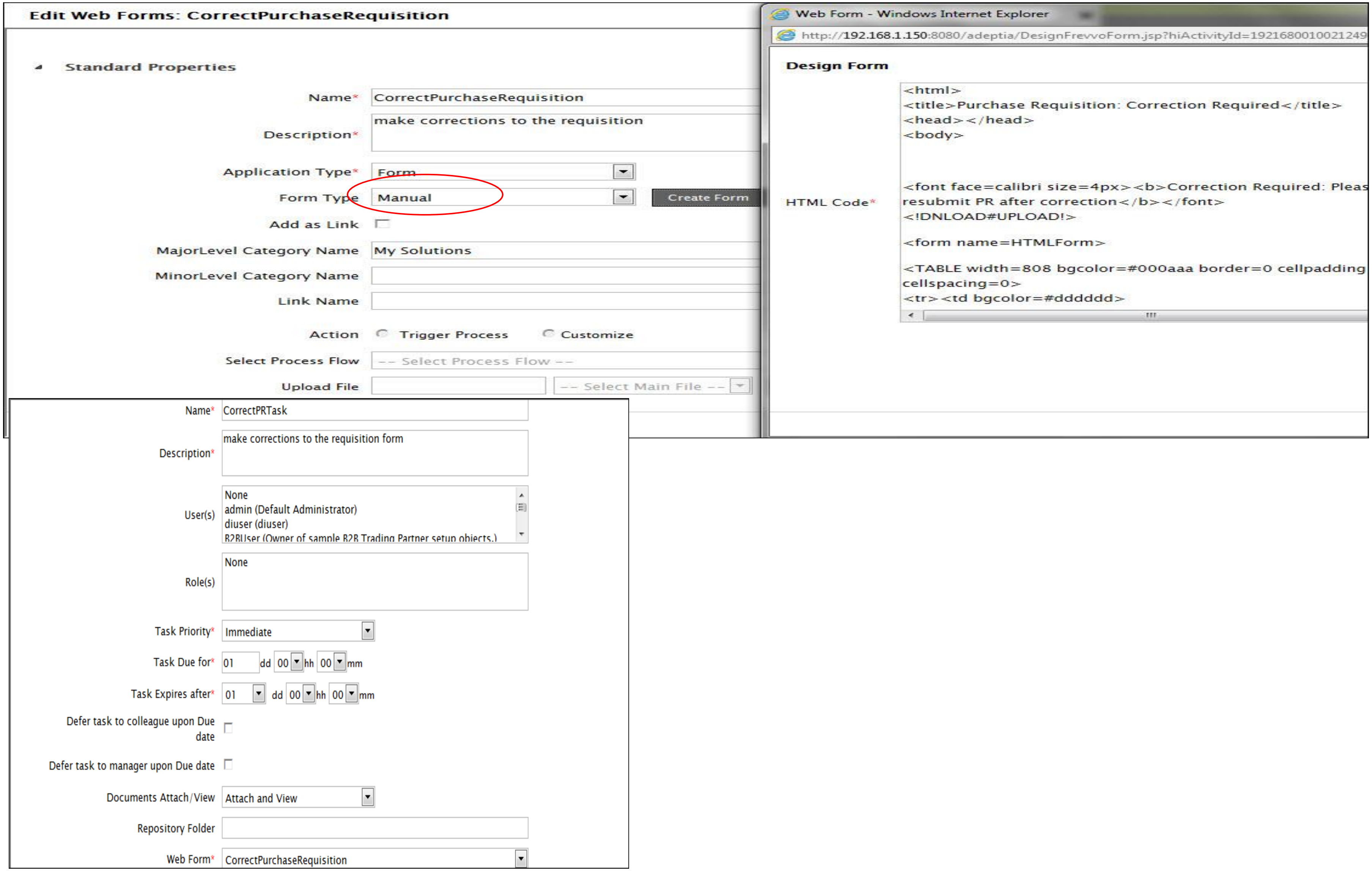
Step 5: Create Review PR Workflow Task



Edit Human WorkFlow: ReviewPRTask

Name*	ReviewPRTask
Description*	review by supervisor
User(s)	demo_manager (demo manager) demo_supervisor (demo supervisor) demouser (Demo User) EDIUser (Owner of sample EDI Trading Partner setup objects.)
Role(s)	None
Task Priority*	Immediate
Task Due for*	01 dd 00 hh 00 mm
Task Expires after*	Never dd 00 hh 00 mm
Defer task to colleague upon Due date	□
Defer task to manager upon Due date	□
Documents Attach/View	Attach and View
Repository Folder	
Web Form*	ReviewPurchaseRequisition

Step 6: Create Correction PR Web Form (and attach it to Workflow task)



Edit Web Forms: CorrectPurchaseRequisition

Standard Properties

Name* CorrectPurchaseRequisition

Description* make corrections to the requisition

Application Type* Form

Form Type Manual

Create Form

Add as Link □

MajorLevel Category Name My Solutions

MinorLevel Category Name

Link Name

Action ○ Trigger Process ○ Customize

Select Process Flow -- Select Process Flow --

Upload File -- Select Main File --

Web Form - Windows Internet Explorer

http://192.168.1.150:8080/adeptia/DesignFrevvoForm.jsp?hiActivityId=1921680010021249

Design Form

HTML Code*

```
<html>
<title>Purchase Requisition: Correction Required</title>
<head></head>
<body>


<font face=calibri size=4px><b>Correction Required: Pleas
resubmit PR after correction</b></font>
<!DNLOAD#UPLOAD!>

<form name=HTMLForm>

<TABLE width=808 bgcolor=#000aaa border=0 cellpadding
cellspacing=0>
<tr><td bgcolor=#dddddd>
```

Name* CorrectPRTask

Description* make corrections to the requisition form

User(s)
None
admin (Default Administrator)
diuser (diuser)
B2BUser (Owner of sample B2B Trading Partner setup objects.)

Role(s) None

Task Priority* Immediate

Task Due for* 01 dd 00 hh 00 mm

Task Expires after* 01 dd 00 hh 00 mm

Defer task to colleague upon Due date □

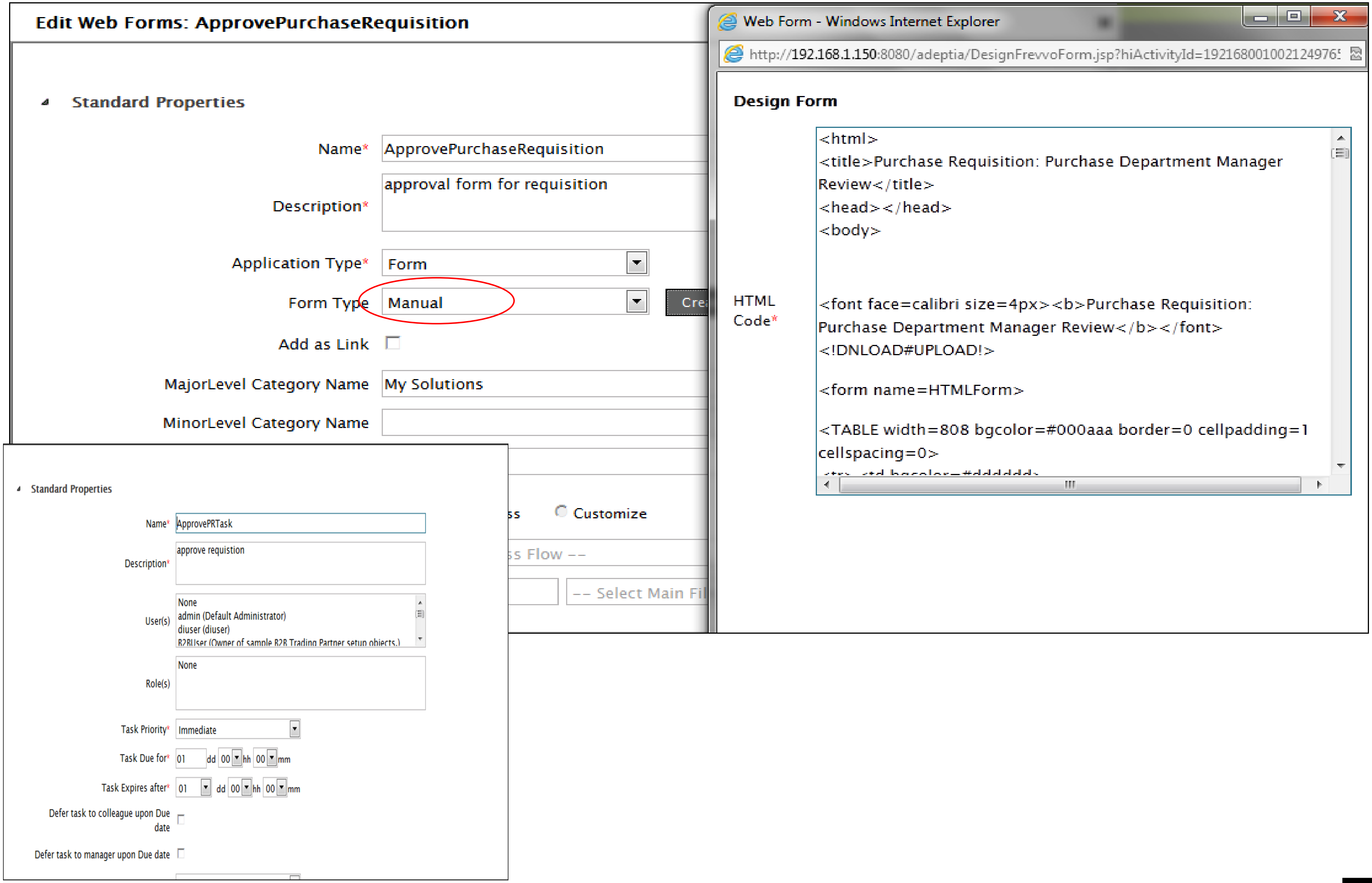
Defer task to manager upon Due date □

Documents Attach/View Attach and View

Repository Folder

Web Form* CorrectPurchaseRequisition

Step 7: Create Approve PR Web Form (and attach it to Workflow task)



Edit Web Forms: ApprovePurchaseRequisition

Standard Properties

Name* ApprovePurchaseRequisition

Description* approval form for requisition

Application Type* Form

Form Type Manual

Add as Link □

MajorLevel Category Name My Solutions

MinorLevel Category Name

○ Customize

-- Select Main Fil

Web Form - Windows Internet Explorer

http://192.168.1.150:8080/adeptia/DesignFrevvoForm.jsp?hiActivityId=19216800100212497б

Design Form

HTML Code*

```
<html>
<title>Purchase Requisition: Purchase Department Manager
Review</title>
<head></head>
<body>


<font face=calibri size=4px><b>Purchase Requisition:
Purchase Department Manager Review</b></font>
<!DNLOAD#UPLOAD!>

<form name=HTMLForm>

<TABLE width=808 bgcolor=#000aaa border=0 cellpadding=1
cellspacing=0>
```

Standard Properties

Name* ApprovePRTask

Description* approve requistion

User(s) None
admin (Default Administrator)
diuser (diuser)
B2BUser (Owner of sample B2B Trading Partner setup objects.)

Role(s) None

Task Priority* Immediate

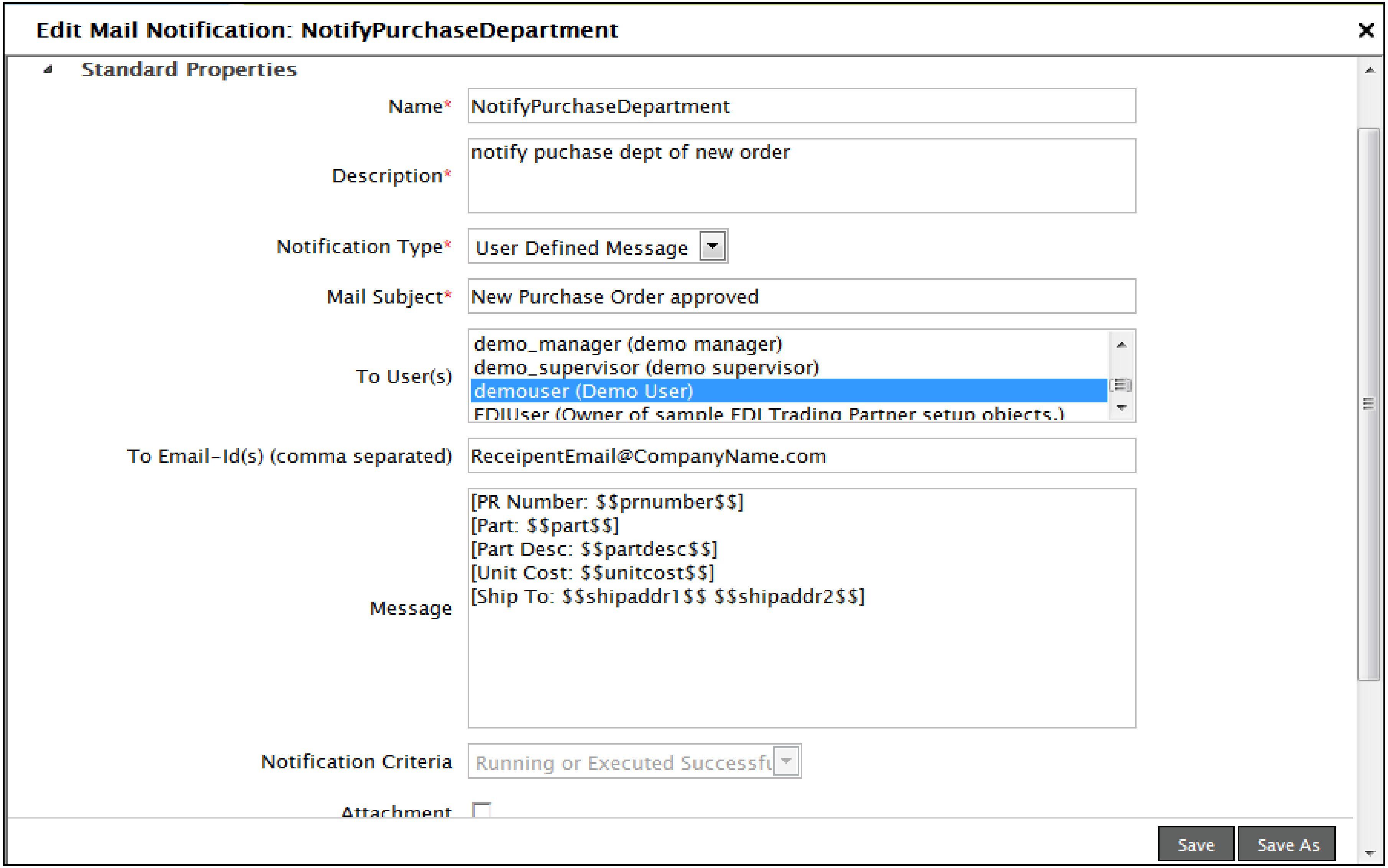
Task Due for* 01 dd 00 hh 00 mm

Task Expires after* 01 dd 00 hh 00 mm

Defer task to colleague upon Due date □

Defer task to manager upon Due date □

Step 8: Create Email Notification



Edit Mail Notification: NotifyPurchaseDepartment

Standard Properties

Name*: NotifyPurchaseDepartment

Description*: notify puchase dept of new order

Notification Type*: User Defined Message

Mail Subject*: New Purchase Order approved

To User(s):
- demo_manager (demo manager)
- demo_supervisor (demo supervisor)
- demouser (Demo User)
- EDIUser (Owner of sample EDI Trading Partner setup objects.)

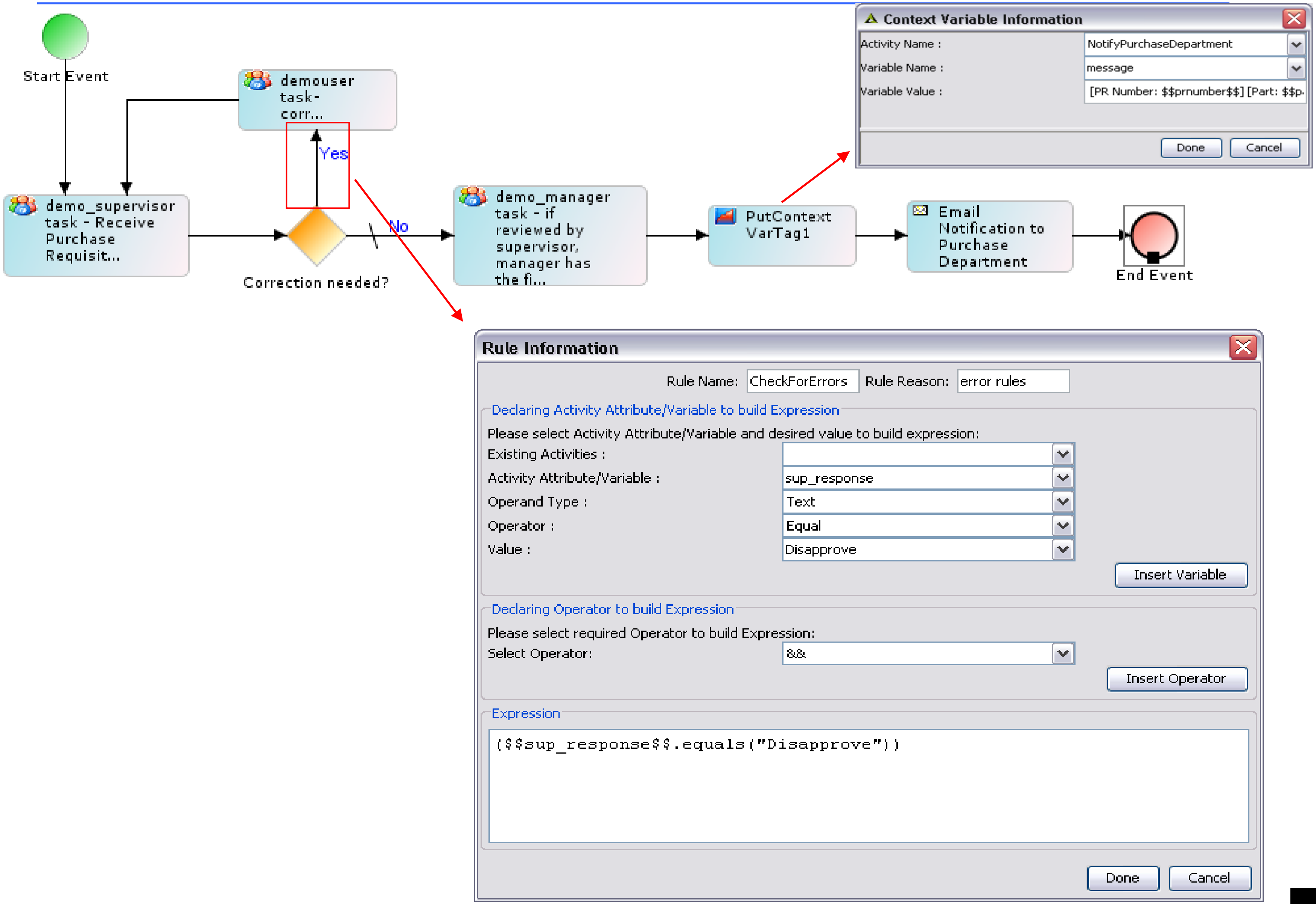
To Email-Id(s) (comma separated): ReceipentEmail@CompanyName.com

Message:
```
[PR Number: $$prnumber$$]
[Part: $$part$$]
[Part Desc: $$partdesc$$]
[Unit Cost: $$unitcost$$]
[Ship To: $$shipaddr1$$ $$shipaddr2$$]
```

Notification Criteria: Running or Executed Successfu

Attachment □

Step 9: Configure Purchase Requisition Workflow


Start Event
demouser task-corr...
Yes
No
demo_supervisor task - Receive Purchase Requisit...
Correction needed?
demo_manager task - if reviewed by supervisor, manager has the fi...
PutContext VarTag1
Email Notification to Purchase Department
End Event

Context Variable Information

Activity Name : NotifyPurchaseDepartment

Variable Name : message

Variable Value : [PR Number: $$prnumber$$] [Part: $$p

Rule Information

Rule Name: CheckForErrors Rule Reason: error rules

Declaring Activity Attribute/Variable to build Expression

Please select Activity Attribute/Variable and desired value to build expression:

Existing Activities :

Activity Attribute/Variable : sup_response

Operand Type : Text

Operator : Equal

Value : Disapprove

Declaring Operator to build Expression

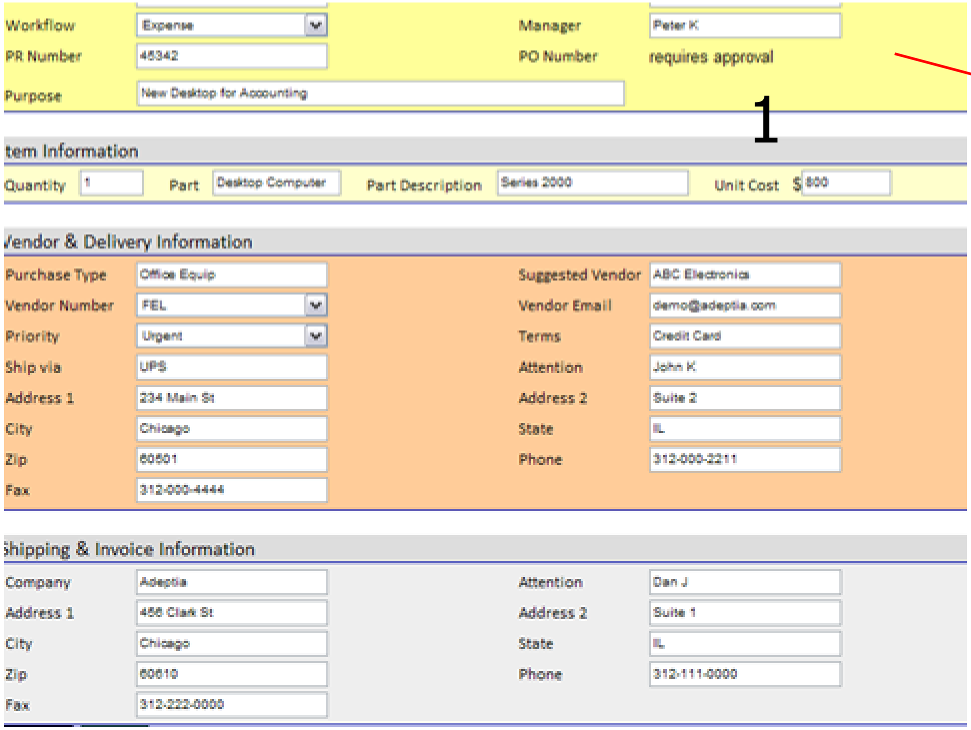
Please select required Operator to build Expression:

Select Operator: &&

Expression

```
($$sup_response$$.equals("Disapprove"))
```

Step 10: Submit Purchase Requisition and track the workflow



1

Workflow	Expense	Manager	Peter K
PR Number	45342	PO Number	requires approval
Purpose	New Desktop for Accounting		

tem Information

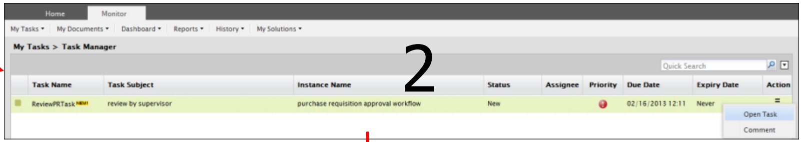
Quantity	1	Part	Desktop Computer	Part Description	Series 2000	Unit Cost	$ 800

/endor & Delivery Information

Purchase Type	Office Equip	Suggested Vendor	ABC Electronics
Vendor Number	FEL	Vendor Email	demo@adeptia.com
Priority	Urgent	Terms	Credit Card
Ship via	UPS	Attention	John K
Address 1	234 Main St	Address 2	Suite 2
City	Chicago	State	IL
Zip	60501	Phone	312-000-2211
Fax	312-000-4444		

hipping & Invoice Information

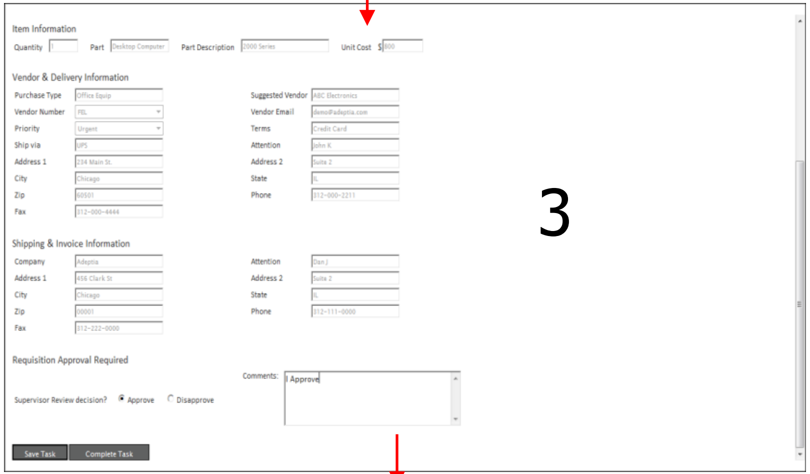
Company	Adeptia	Attention	Dan J
Address 1	456 Clark St	Address 2	Suite 1
City	Chicago	State	IL
Zip	60610	Phone	312-111-0000
Fax	312-222-0000		



2

My Tasks > Task Manager

	Task Name	Task Subject	Instance Name	Status	Assignee	Priority	Due Date	Expiry Date	Action
■	ReviewPRTask NEW!	review by supervisor	purchase requisition approval workflow	New		●	02/16/2013 12:11	Never	≡



3

Item Information

Quantity	1	Part	Desktop Computer	Part Description	2000 Series	Unit Cost	$ 800

Vendor & Delivery Information

Purchase Type	Office Equip	Suggested Vendor	ABC Electronics
Vendor Number	FEL	Vendor Email	demo@adeptia.com
Priority	Urgent	Terms	Credit Card
Ship via	UPS	Attention	John K
Address 1	234 Main St.	Address 2	Suite 2
City	Chicago	State	IL
Zip	60501	Phone	312-000-2211
Fax	312-000-4444		

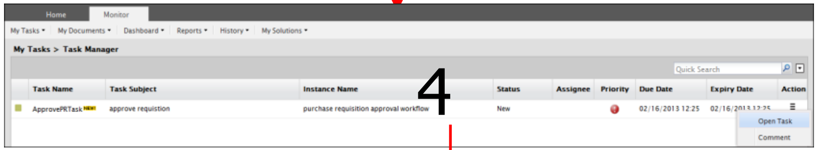
Shipping & Invoice Information

Company	Adeptia	Attention	Dan J
Address 1	456 Clark St	Address 2	Suite 2
City	Chicago	State	IL
Zip	00001	Phone	312-111-0000
Fax	312-222-0000		

Requisition Approval Required

Supervisor Review decision? ● Approve ○ Disapprove

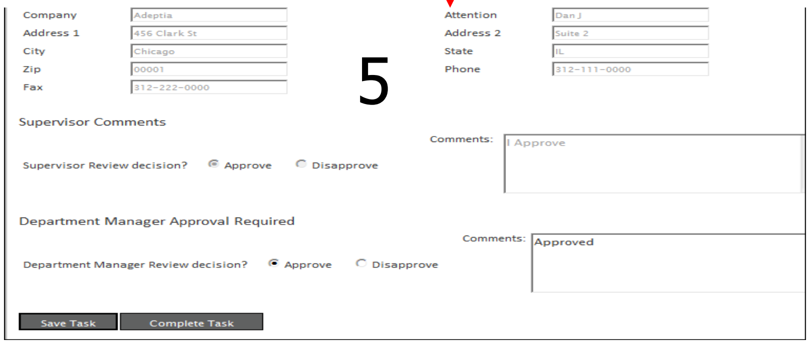
Comments: I Approve



4

My Tasks > Task Manager

	Task Name	Task Subject	Instance Name	Status	Assignee	Priority	Due Date	Expiry Date	Action
■	ApprovePRTask NEW!	approve requistion	purchase requisition approval workflow	New		●	02/16/2013 12:25	02/16/2013 12:25	≡



5

Company	Adeptia	Attention	Dan J
Address 1	456 Clark St	Address 2	Suite 2
City	Chicago	State	IL
Zip	00001	Phone	312-111-0000
Fax	312-222-0000		

Supervisor Comments

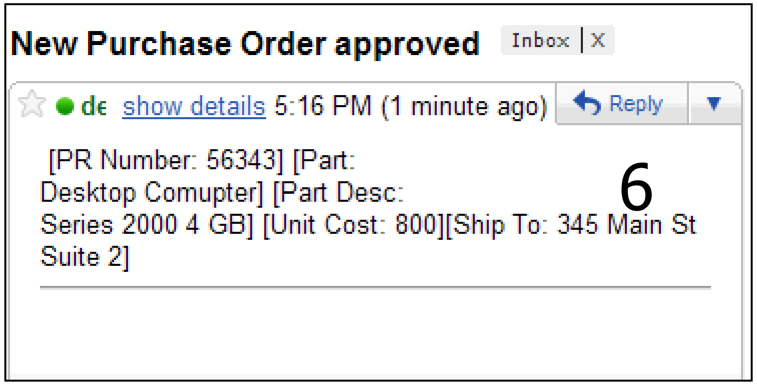
Supervisor Review decision? ● Approve ○ Disapprove

Comments: I Approve

Department Manager Approval Required

Department Manager Review decision? ● Approve ○ Disapprove

Comments: Approved



6

New Purchase Order approved

5:16 PM (1 minute ago)

[PR Number: 56343] [Part: Desktop Comupter] [Part Desc: Series 2000 4 GB] [Unit Cost: 800][Ship To: 345 Main St Suite 2]

Training schedule

Day 1

Morning session
9am to 12pm, 10 min break at 10:30am

- Introduction
  - Overview of Adeptia Suite
  - Architecture
  - Project approach
- Process Modeler & Simulation (BPM)
  - Modeler
  - Simulator
- Process Designer
- BPM Exercise – Purchase Requisition Workflow (BPM)
  - Web Forms
  - Workflow

Afternoon session
1pm to 4:30pm, 10 min break @ 2:20pm and 3:20pm

- Integration/ETL Exercises:
  - Employee Benefits
  - Purchase Order
  - Inventory Items
  - Order Fulfillment
- Services: Description and Exercises
  - Sources
  - Targets
  - Data Dictionary
  - Schema
  - Mapping
  - Web Services
  - Workflow
- Events & Triggers

Day 2

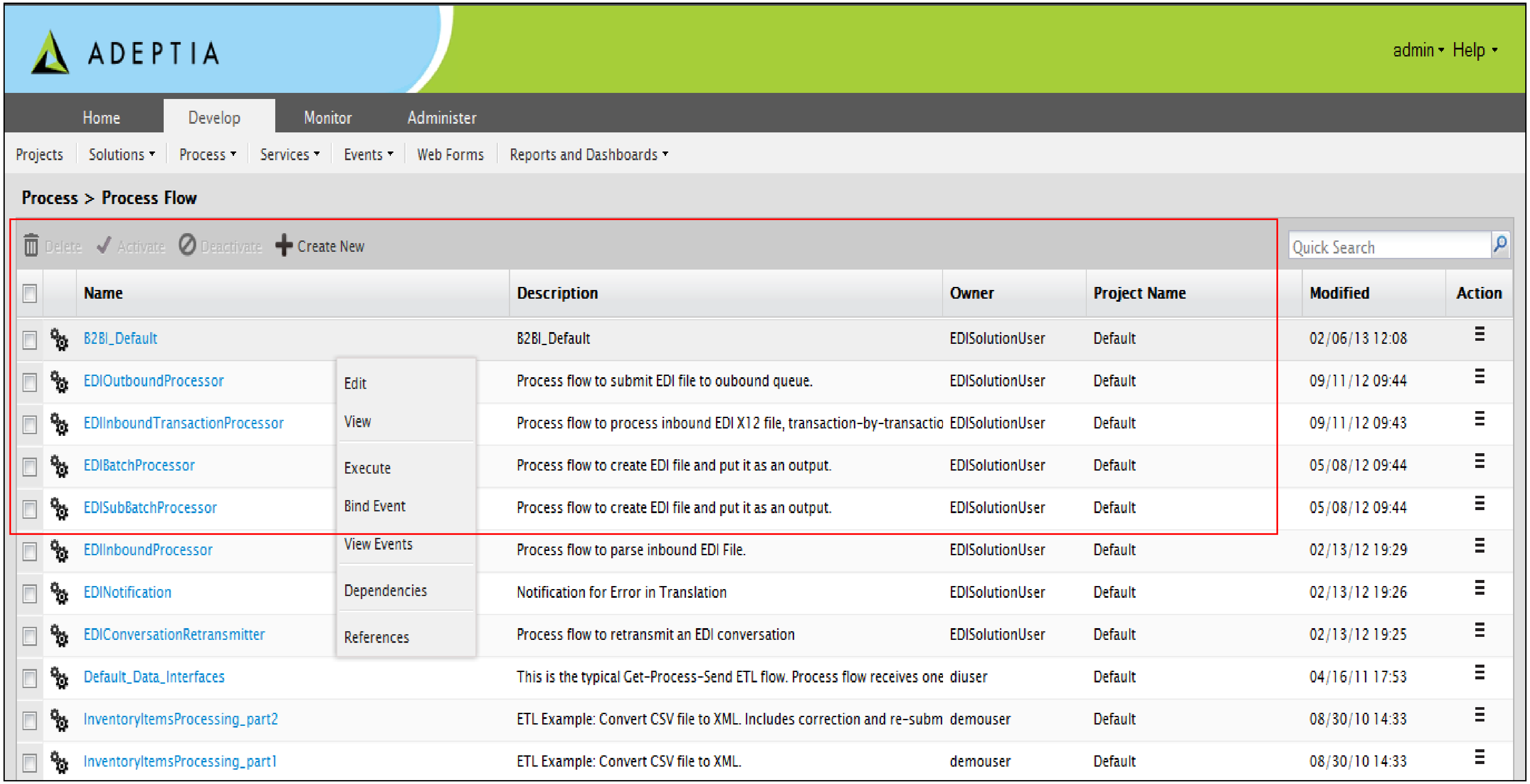
Morning session
9am to 12pm, 10 min break at 10:30am

- Advanced Mapping & Process Designer functions
  - Process Designer
    - Decision nodes
    - Context Variables
    - Sub Processes
  - Mapping
    - Math, String and Conditional functions
    - Database lookups
    - Context variables
    - Multiple Source and Target mapping
- Execute flow
  - Trigger
  - Logs
  - Reports

Afternoon session
1pm to 4pm, 10 min break @ 2:30pm

- Managing Solutions
  - My Tasks and Task History
  - Solution and Performance Monitoring
  - Logs
- Administration
  - User management
  - Configuration and Properties
  - System monitoring
  - Maintenance activities

Training Exercises focused on Integration Suite



Process > Process Flow

□		Name	Description	Owner	Project Name	Modified	Action
□	⚙	B2BI_Default	B2BI_Default	EDISolutionUser	Default	02/06/13 12:08	≡
□	⚙	EDIOutboundProcessor	Process flow to submit EDI file to oubound queue.	EDISolutionUser	Default	09/11/12 09:44	≡
□	⚙	EDIInboundTransactionProcessor	Process flow to process inbound EDI X12 file, transaction-by-transactio	EDISolutionUser	Default	09/11/12 09:43	≡
□	⚙	EDIBatchProcessor	Process flow to create EDI file and put it as an output.	EDISolutionUser	Default	05/08/12 09:44	≡
□	⚙	EDISubBatchProcessor	Process flow to create EDI file and put it as an output.	EDISolutionUser	Default	05/08/12 09:44	≡
□	⚙	EDIInboundProcessor	Process flow to parse inbound EDI File.	EDISolutionUser	Default	02/13/12 19:29	≡
□	⚙	EDINotification	Notification for Error in Translation	EDISolutionUser	Default	02/13/12 19:26	≡
□	⚙	EDIConversationRetransmitter	Process flow to retransmit an EDI conversation	EDISolutionUser	Default	02/13/12 19:25	≡
□	⚙	Default_Data_Interfaces	This is the typical Get-Process-Send ETL flow. Process flow receives one	diuser	Default	04/16/11 17:53	≡
□	⚙	InventoryItemsProcessing_part2	ETL Example: Convert CSV file to XML. Includes correction and re-subm	demouser	Default	08/30/10 14:33	≡
□	⚙	InventoryItemsProcessing_part1	ETL Example: Convert CSV file to XML.	demouser	Default	08/30/10 14:33	≡

Exercise: Employee Benefits Data Conversion

Exercise: Employee Benefits Data Conversion

This process flow receives Employee Benefits data in Excel files and converts it into XML output files.

Design:

- Design this process in the Process Designer using a top-down approach

Configure:

- Configure the process by creating below activities and attaching them to the appropriate steps in the process flow:
  - Create File Source activity
  - Create Excel Data Schema activity
  - Create XML Data Schema activity
  - Create Excel to XML Mapping activity
  - Use PutContextVar action in the process flow to dynamically name output files
  - In the process flow set File Source activity’s ‘eventContextEnabled’ property to True

Deploy:

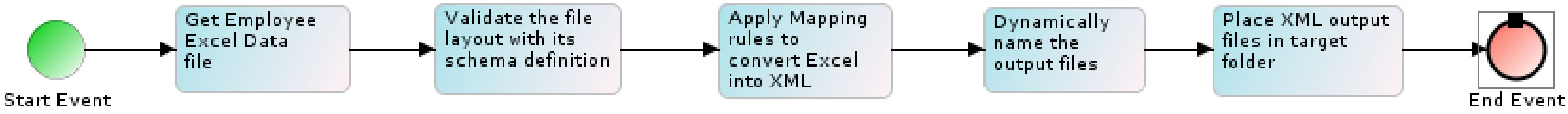
- Create a File Event
- Register the File Event with Employee Benefits process flow
- Activate File Event

Exercise: Design Employee Benefits Data Conversion process flow

Using the Top-down approach we will first design our process flow as shown here.

Design consists of following steps :

1. Get the employee benefits excel files from a source folder
2. Validate the data files against its schema
3. Convert the data file into XML
4. Name the XML file dynamically
5. Place the XML file in target folder


Start Event
Get Employee Excel Data file
Validate the file layout with its schema definition
Apply Mapping rules to convert Excel into XML
Dynamically name the output files
Place XML output files in target folder
End Event

File Source

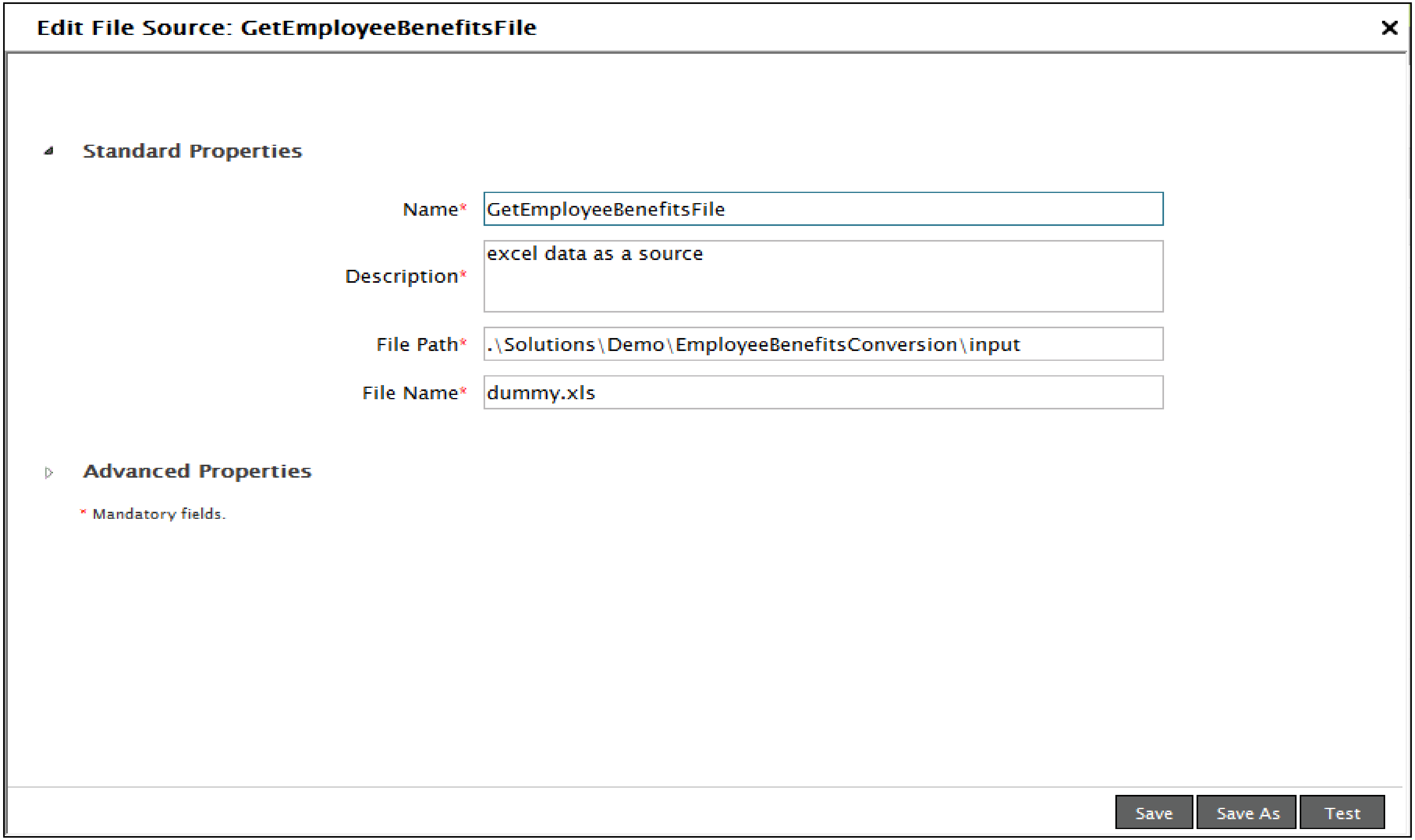
Source Schema

Mapping

Action
(PutContextVariable)

File Target

Exercise: Employee Benefits Data Conversion (Create File Source)



Edit File Source: GetEmployeeBenefitsFile

Standard Properties

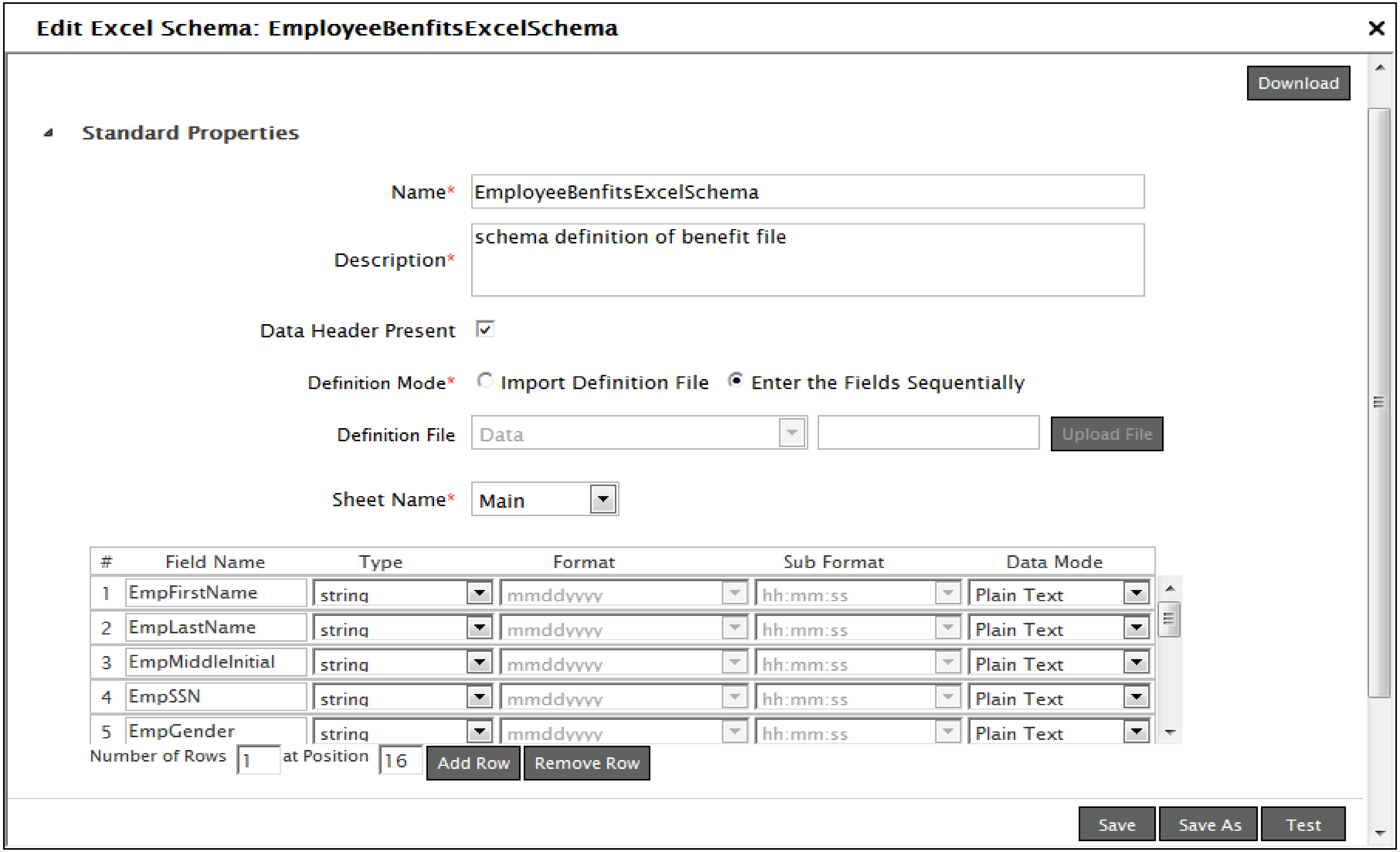
Name*	GetEmployeeBenefitsFile
Description*	excel data as a source
File Path*	.\Solutions\Demo\EmployeeBenefitsConversion\input
File Name*	dummy.xls

Advanced Properties

* Mandatory fields.

Refer to Notes below

Exercise: Employee Benefits Data Conversion (Create Source Schema)



Edit Excel Schema: EmployeeBenfitsExcelSchema

Standard Properties

Name* EmployeeBenfitsExcelSchema

Description* schema definition of benefit file

Data Header Present ☑

Definition Mode* ○ Import Definition File ● Enter the Fields Sequentially

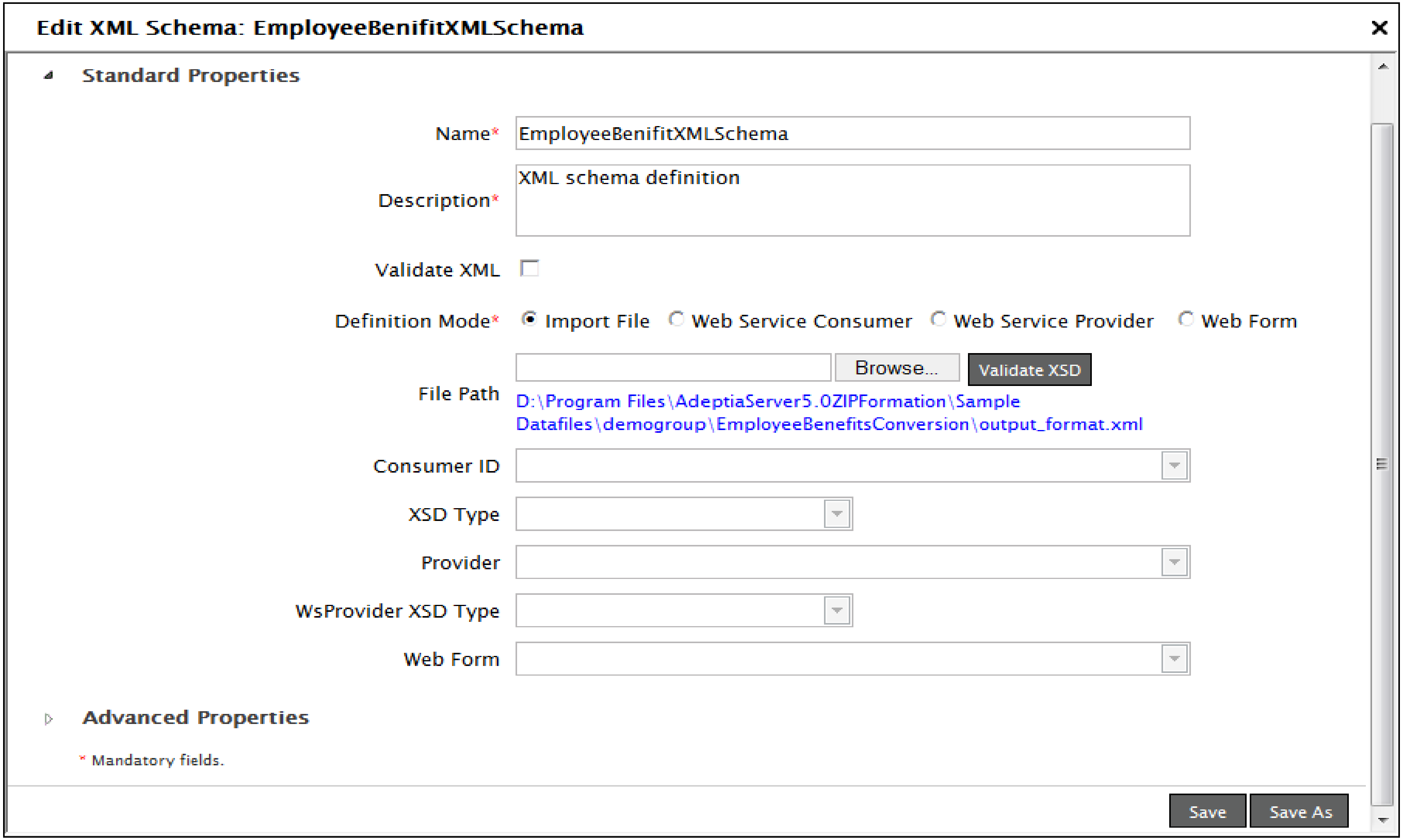
Definition File Data

Sheet Name* Main

#	Field Name	Type	Format	Sub Format	Data Mode
1	EmpFirstName	string	mmddyyyy	hh:mm:ss	Plain Text
2	EmpLastName	string	mmddyyyy	hh:mm:ss	Plain Text
3	EmpMiddleInitial	string	mmddyyyy	hh:mm:ss	Plain Text
4	EmpSSN	string	mmddyyyy	hh:mm:ss	Plain Text
5	EmpGender	string	mmddyyyy	hh:mm:ss	Plain Text

Number of Rows 1 at Position 16

Exercise: Employee Benefits Data Conversion (Create Target Schema)



Edit XML Schema: EmployeeBenifitXMLSchema

Standard Properties

Name*: EmployeeBenifitXMLSchema

Description*: XML schema definition

Validate XML: □

Definition Mode*: ● Import File ○ Web Service Consumer ○ Web Service Provider ○ Web Form

File Path: D:\Program Files\AdeptiaServer5.0ZIPFormation\Sample Datafiles\demogroup\EmployeeBenefitsConversion\output_format.xml

Consumer ID:

XSD Type:

Provider:

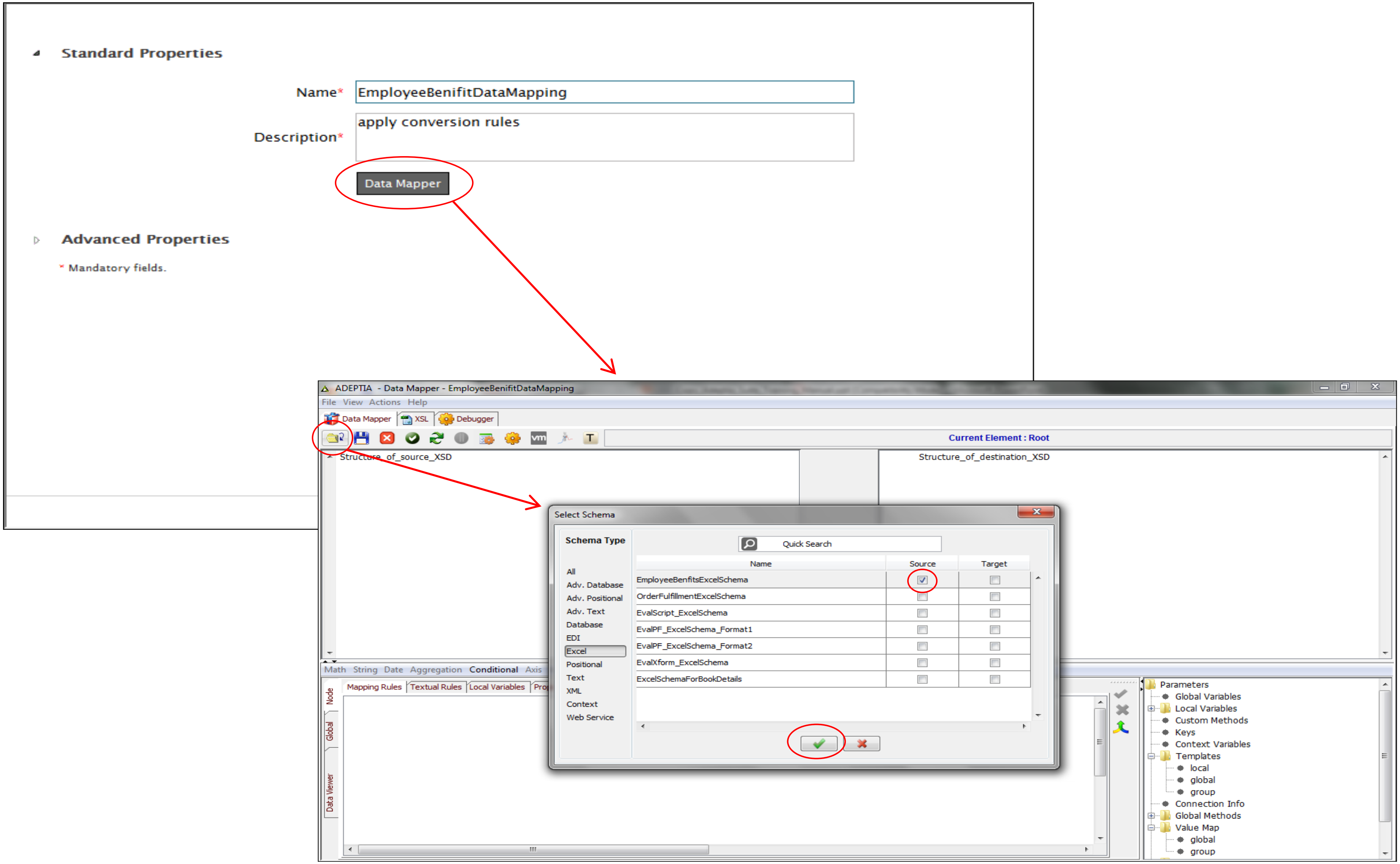
WsProvider XSD Type:

Web Form:

Advanced Properties

* Mandatory fields.

Exercise: Employee Benefits Data Conversion (Create Mapping)



Standard Properties

Name* EmployeeBenifitDataMapping

Description* apply conversion rules

Data Mapper

Advanced Properties

* Mandatory fields.

ADEPTIA - Data Mapper - EmployeeBenifitDataMapping

Current Element : Root

Structure_of_source_XSD

Structure_of_destination_XSD

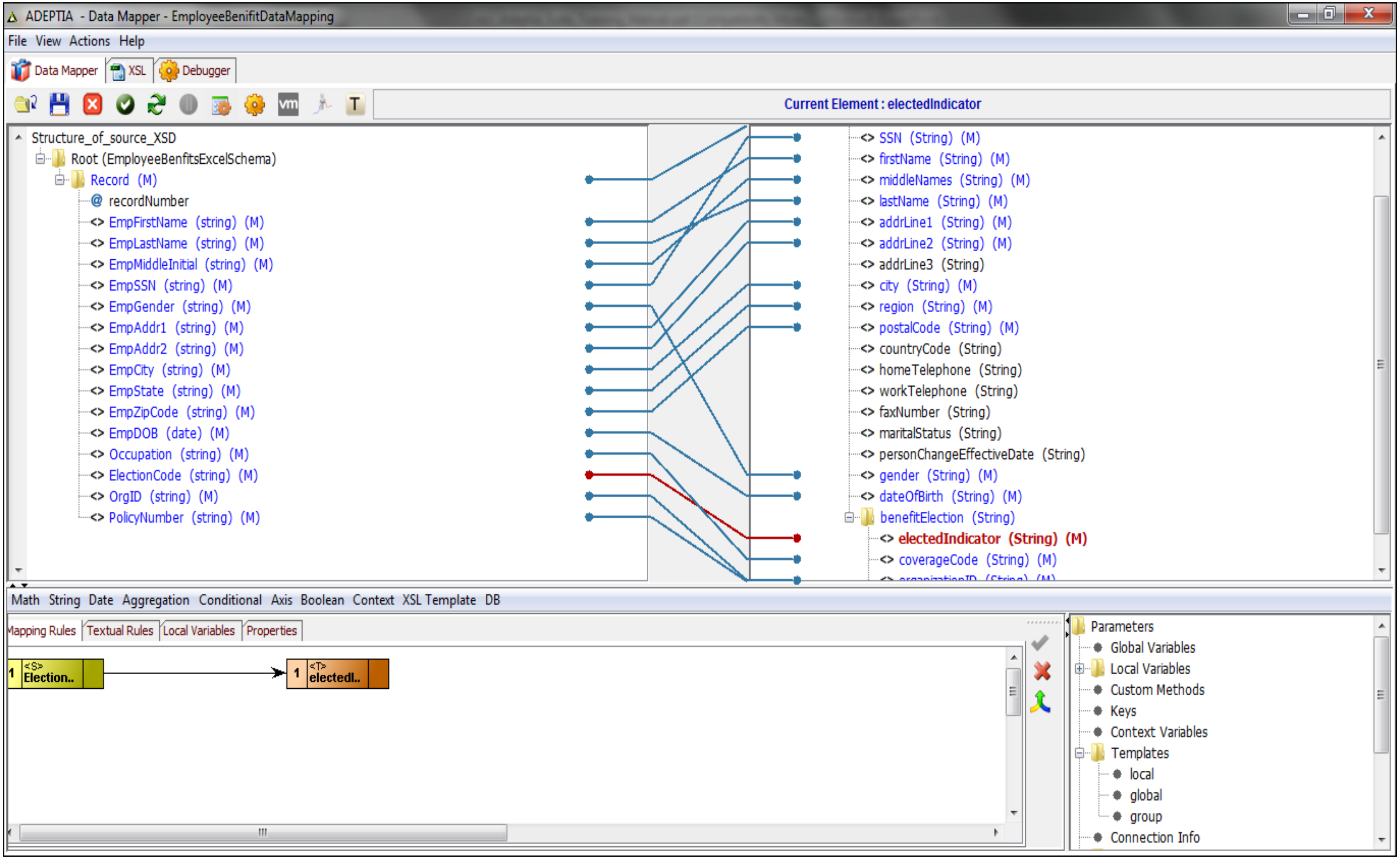
Select Schema

Schema Type: All, Adv. Database, Adv. Positional, Adv. Text, Database, EDI, Excel, Positional, Text, XML, Context, Web Service

Quick Search

Name	Source	Target
EmployeeBenfitsExcelSchema	☑	□
OrderFulfillmentExcelSchema	□	□
EvalScript_ExcelSchema	□	□
EvalPF_ExcelSchema_Format1	□	□
EvalPF_ExcelSchema_Format2	□	□
EvalXform_ExcelSchema	□	□
ExcelSchemaForBookDetails	□	□

Exercise: Employee Benefits Data Conversion (Create Mapping)



ADEPTIA - Data Mapper - EmployeeBenifitDataMapping
File View Actions Help
Data Mapper
XSL
Debugger
Math String Date Aggregation Conditional Axis Boolean Context XSL Template DB
Mapping Rules
Textual Rules
Local Variables
Properties

Current Element : electedIndicator

Structure_of_source_XSD
- Root (EmployeeBenfitsExcelSchema)
  - Record (M)
    - @ recordNumber
    - <> EmpFirstName (string) (M)
    - <> EmpLastName (string) (M)
    - <> EmpMiddleInitial (string) (M)
    - <> EmpSSN (string) (M)
    - <> EmpGender (string) (M)
    - <> EmpAddr1 (string) (M)
    - <> EmpAddr2 (string) (M)
    - <> EmpCity (string) (M)
    - <> EmpState (string) (M)
    - <> EmpZipCode (string) (M)
    - <> EmpDOB (date) (M)
    - <> Occupation (string) (M)
    - <> ElectionCode (string) (M)
    - <> OrgID (string) (M)
    - <> PolicyNumber (string) (M)

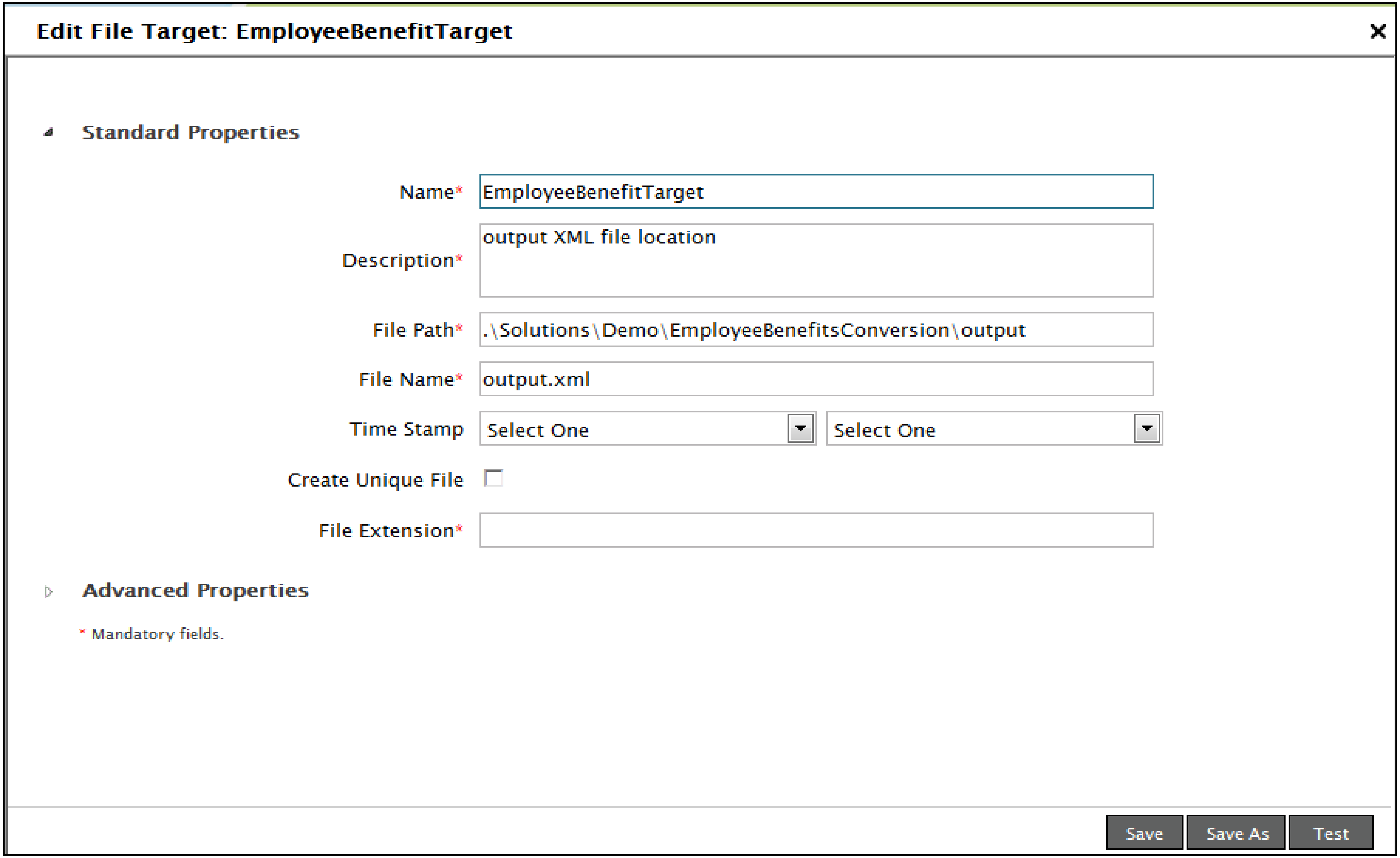
- <> SSN (String) (M)
- <> firstName (String) (M)
- <> middleNames (String) (M)
- <> lastName (String) (M)
- <> addrLine1 (String) (M)
- <> addrLine2 (String) (M)
- <> addrLine3 (String)
- <> city (String) (M)
- <> region (String) (M)
- <> postalCode (String) (M)
- <> countryCode (String)
- <> homeTelephone (String)
- <> workTelephone (String)
- <> faxNumber (String)
- <> maritalStatus (String)
- <> personChangeEffectiveDate (String)
- <> gender (String) (M)
- <> dateOfBirth (String) (M)
- benefitElection (String)
  - <> electedIndicator (String) (M)
  - <> coverageCode (String) (M)
  - <> organizationID (String) (M)

1 <S> Election..

1 <T> electedI..

- Parameters
  - Global Variables
  - Local Variables
  - Custom Methods
  - Keys
  - Context Variables
  - Templates
    - local
    - global
    - group
  - Connection Info

Exercise: Employee Benefits Data Conversion (Create File Target)



Edit File Target: EmployeeBenefitTarget

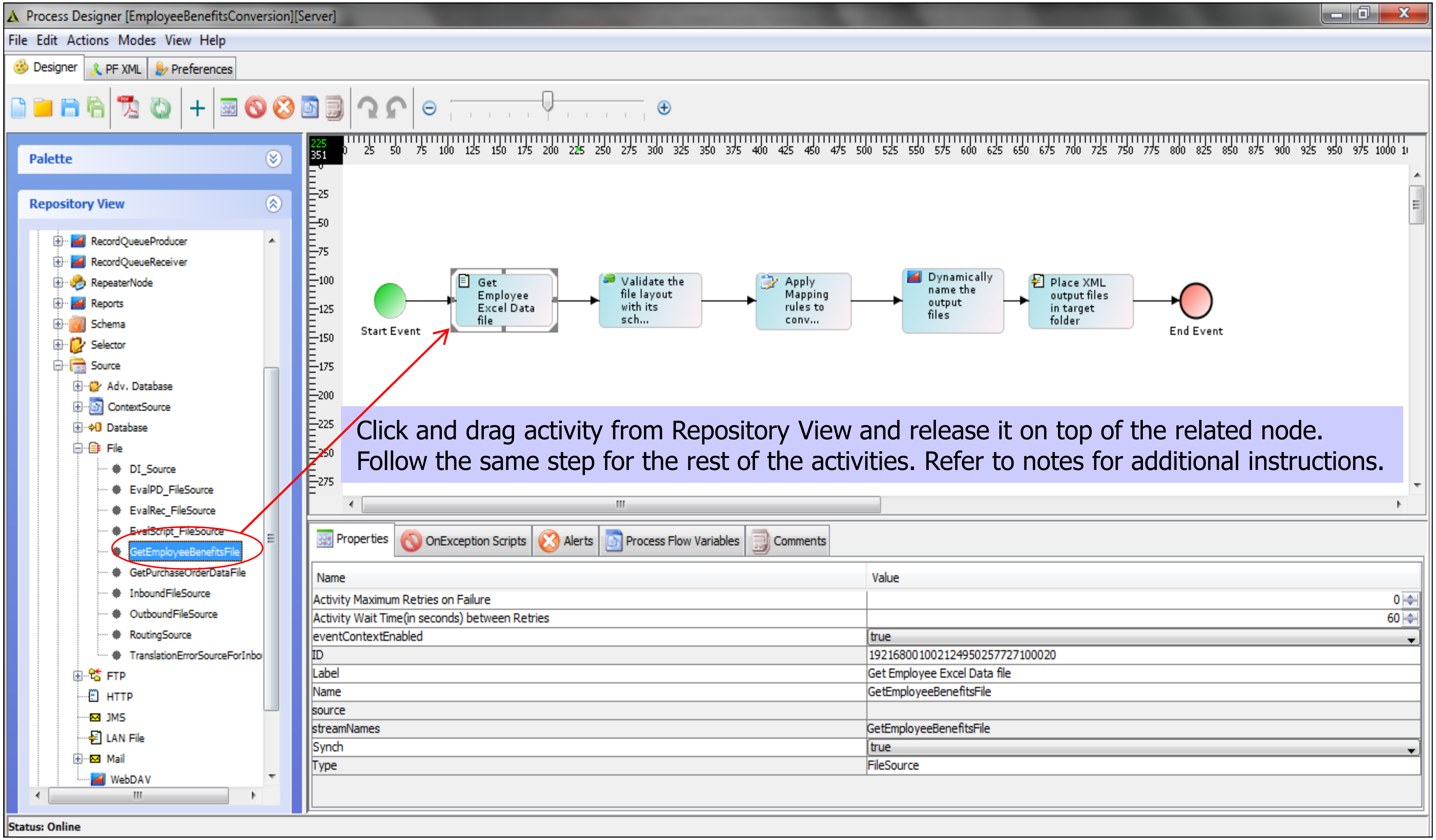
Standard Properties

Name*	EmployeeBenefitTarget
Description*	output XML file location
File Path*	.\Solutions\Demo\EmployeeBenefitsConversion\output
File Name*	output.xml
Time Stamp	Select One ▾ Select One ▾
Create Unique File	□
File Extension*	

Advanced Properties

* Mandatory fields.

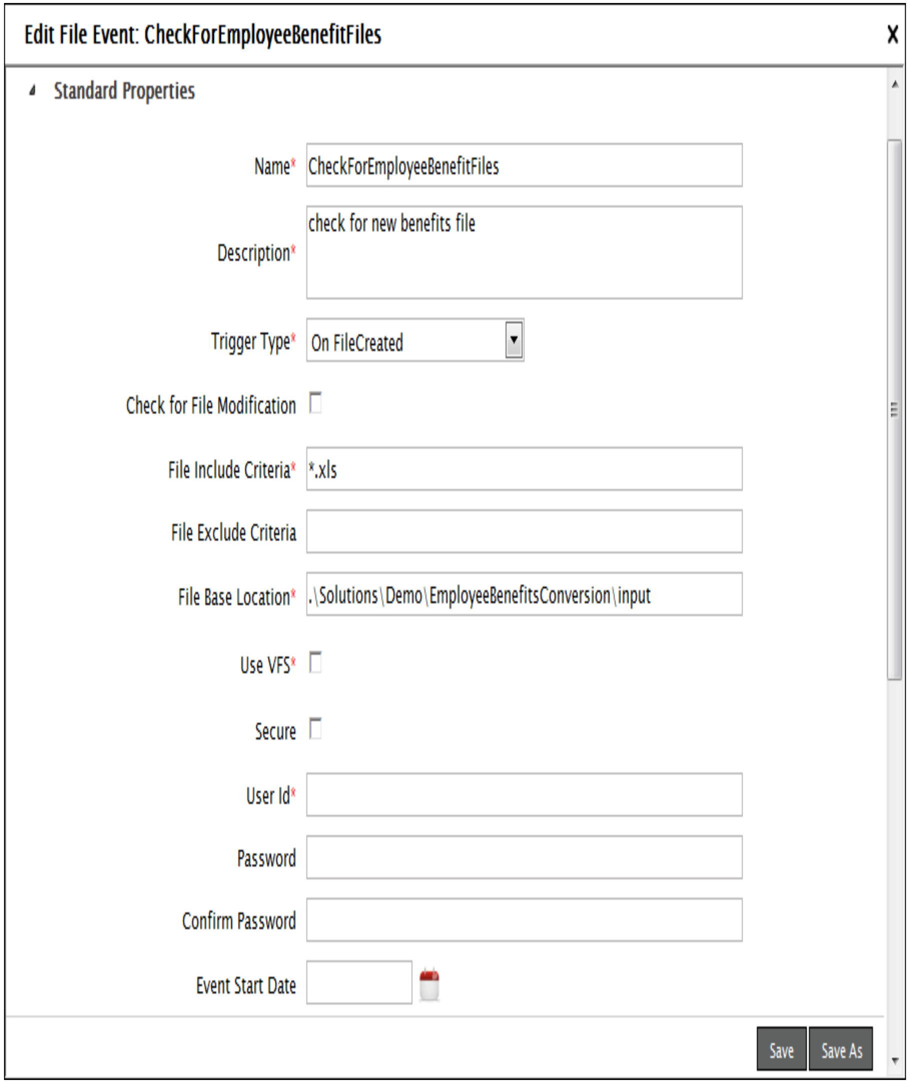
Exercise: Employee Benefits Data Conversion (configure process flow)



Click and drag activity from Repository View and release it on top of the related node. Follow the same step for the rest of the activities. Refer to notes for additional instructions.

Name	Value
Activity Maximum Retries on Failure	0
Activity Wait Time(in seconds) between Retries	60
eventContextEnabled	true
ID	192168001002124950257727100020
Label	Get Employee Excel Data file
Name	GetEmployeeBenefitsFile
source	
streamNames	GetEmployeeBenefitsFile
Synch	true
Type	FileSource

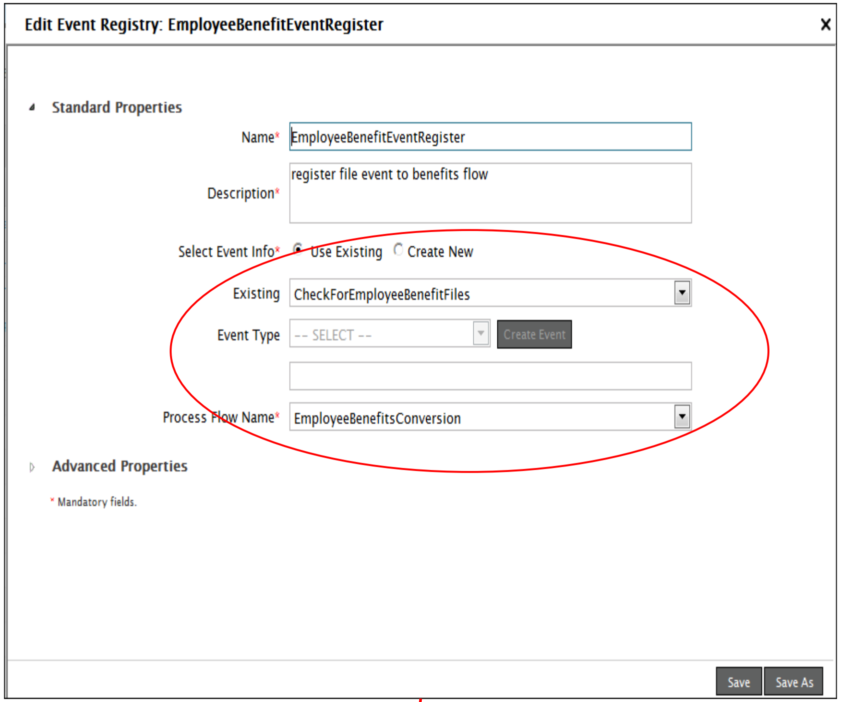
Exercise: Employee Benefits Data Conversion (deploy process flow)



Edit File Event: CheckForEmployeeBenefitFiles X

Standard Properties

Name*	CheckForEmployeeBenefitFiles
Description*	check for new benefits file
Trigger Type*	On FileCreated
Check for File Modification	☐
File Include Criteria*	*.xls
File Exclude Criteria	
File Base Location*	.\Solutions\Demo\EmployeeBenefitsConversion\input
Use VFS*	☐
Secure	☐
User Id*	
Password	
Confirm Password	
Event Start Date	



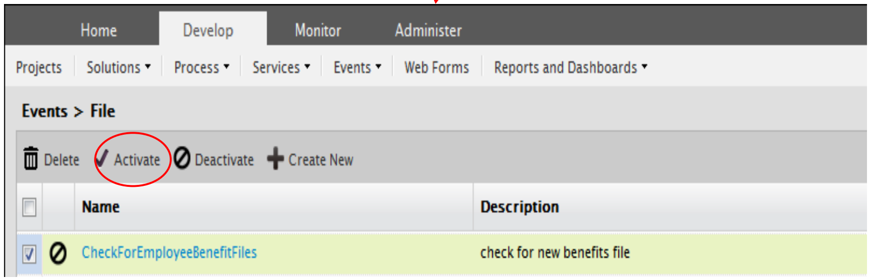
Edit Event Registry: EmployeeBenefitEventRegister x

Standard Properties

Name*	EmployeeBenefitEventRegister
Description*	register file event to benefits flow
Select Event Info*	● Use Existing ○ Create New
Existing	CheckForEmployeeBenefitFiles
Event Type	-- SELECT --
Process Flow Name*	EmployeeBenefitsConversion

Advanced Properties

* Mandatory fields.



Events > File

☐		Name	Description
☑	⊘	CheckForEmployeeBenefitFiles	check for new benefits file

Exercise: Purchase Order Processing

Exercise: Purchase Order Processing

This process flow integrates Purchase Order data into a back-end Point of Sales Database.

Design:

- Design this process in the Process Designer using a top-down approach

Configure:

- Configure the process by creating below activities and attaching them to the appropriate steps in the process flow:
  - Create File Source activity
  - Create XML Data Schema activity
  - Create Advanced Database Schema activity
  - Create XML to Adv Database Mapping activity

Deploy:

- Create a File Event
- Register the File Event with Purchase Order Processing process flow
- Activate File Event

Prerequisites for this process:

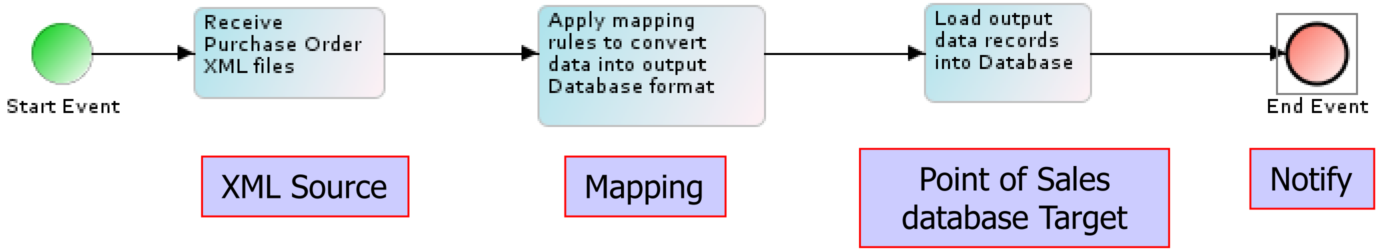
- Create a database named ‘CentralPOSDb’ in any Relational Database instance (such as MS Sql Server, MySQL, Oracle etc)
- Run 3 table scripts in the database (scripts are located in install directory’s demogroup/PurchaseOrderProcessing)
- In Adeptia, go to Administer > Database Info and edit the object related to this Database and provide correct connection info

Exercise: Design Purchase Order Processing flow

Using the Top-down approach we will first design our process flow as shown here.

Design consists of following steps :

1. Get the Purchase Order files from a source folder
2. Convert the data file into Database output
3. Insert data into Database
4. Notify when the process finishes


Start Event
Receive Purchase Order XML files
Apply mapping rules to convert data into output Database format
Load output data records into Database
End Event

XML Source

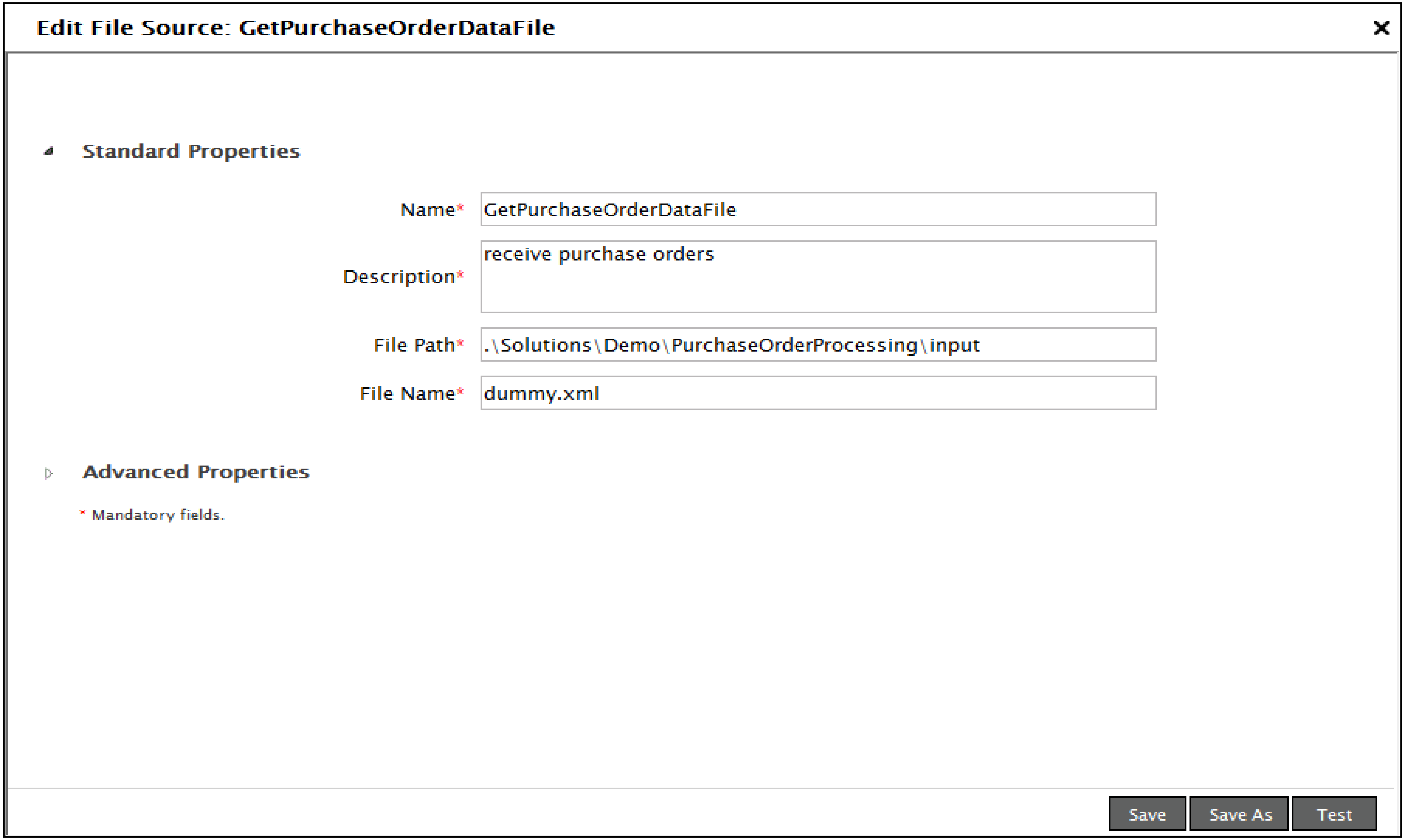
Mapping

Point of Sales database Target

Notify

Note: For data types related to XML and Database, there is no need to specify their schema activities in the process flow design. The schemas are already included in their Source/Target definitions and thus explicitly defining them in the flow design is not required.

Exercise: Purchase Order Processing (Create File Source)



Edit File Source: GetPurchaseOrderDataFile

Standard Properties

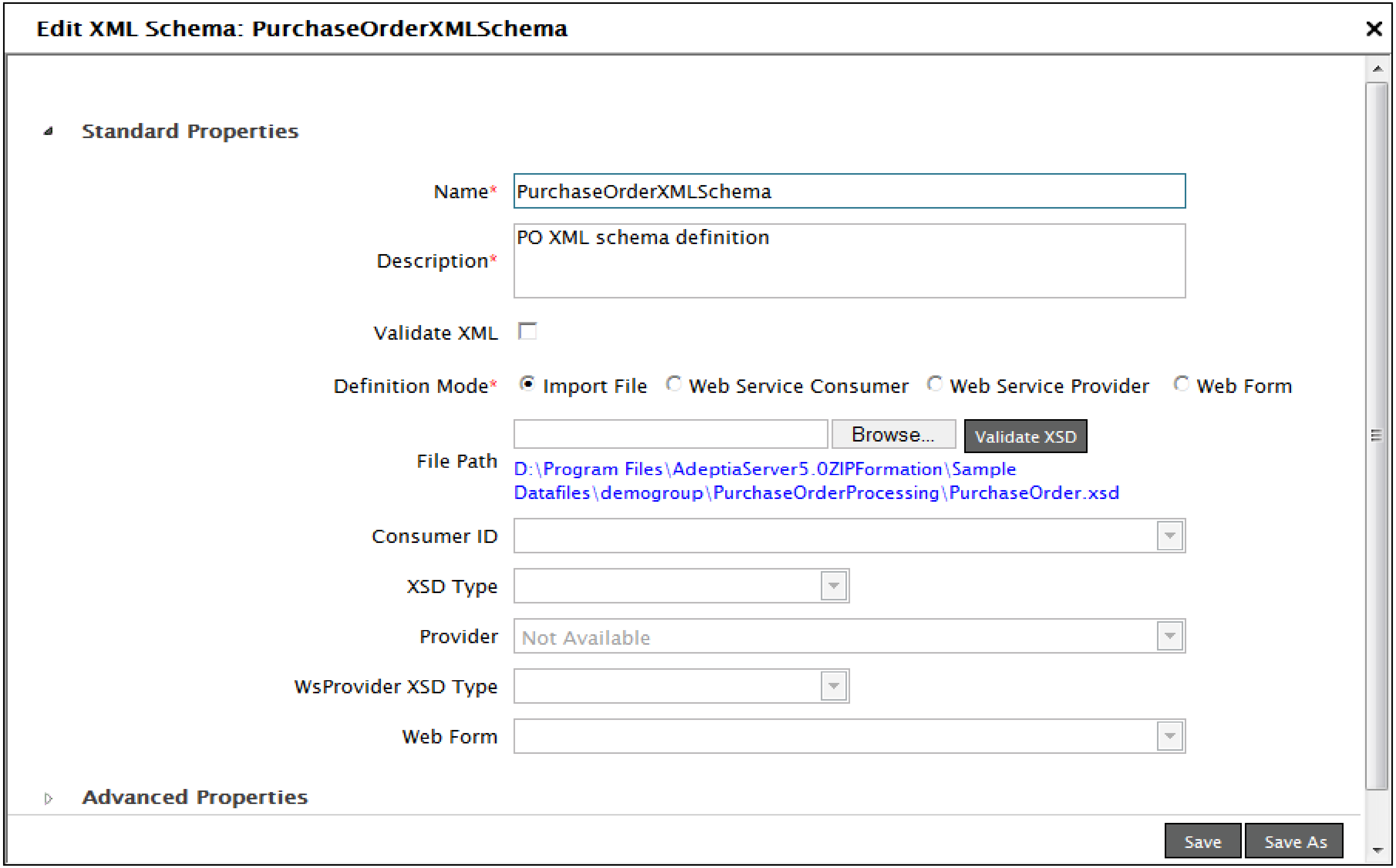
Name*	GetPurchaseOrderDataFile
Description*	receive purchase orders
File Path*	.\Solutions\Demo\PurchaseOrderProcessing\input
File Name*	dummy.xml

Advanced Properties

* Mandatory fields.

Refer to Notes below

Exercise: Purchase Order Processing (Create XML Schema)



Edit XML Schema: PurchaseOrderXMLSchema

Standard Properties

Name*: PurchaseOrderXMLSchema

Description*: PO XML schema definition

Validate XML: □

Definition Mode*: ● Import File ○ Web Service Consumer ○ Web Service Provider ○ Web Form

File Path: D:\Program Files\AdeptiaServer5.0ZIPFormation\Sample Datafiles\demogroup\PurchaseOrderProcessing\PurchaseOrder.xsd

Consumer ID:

XSD Type:

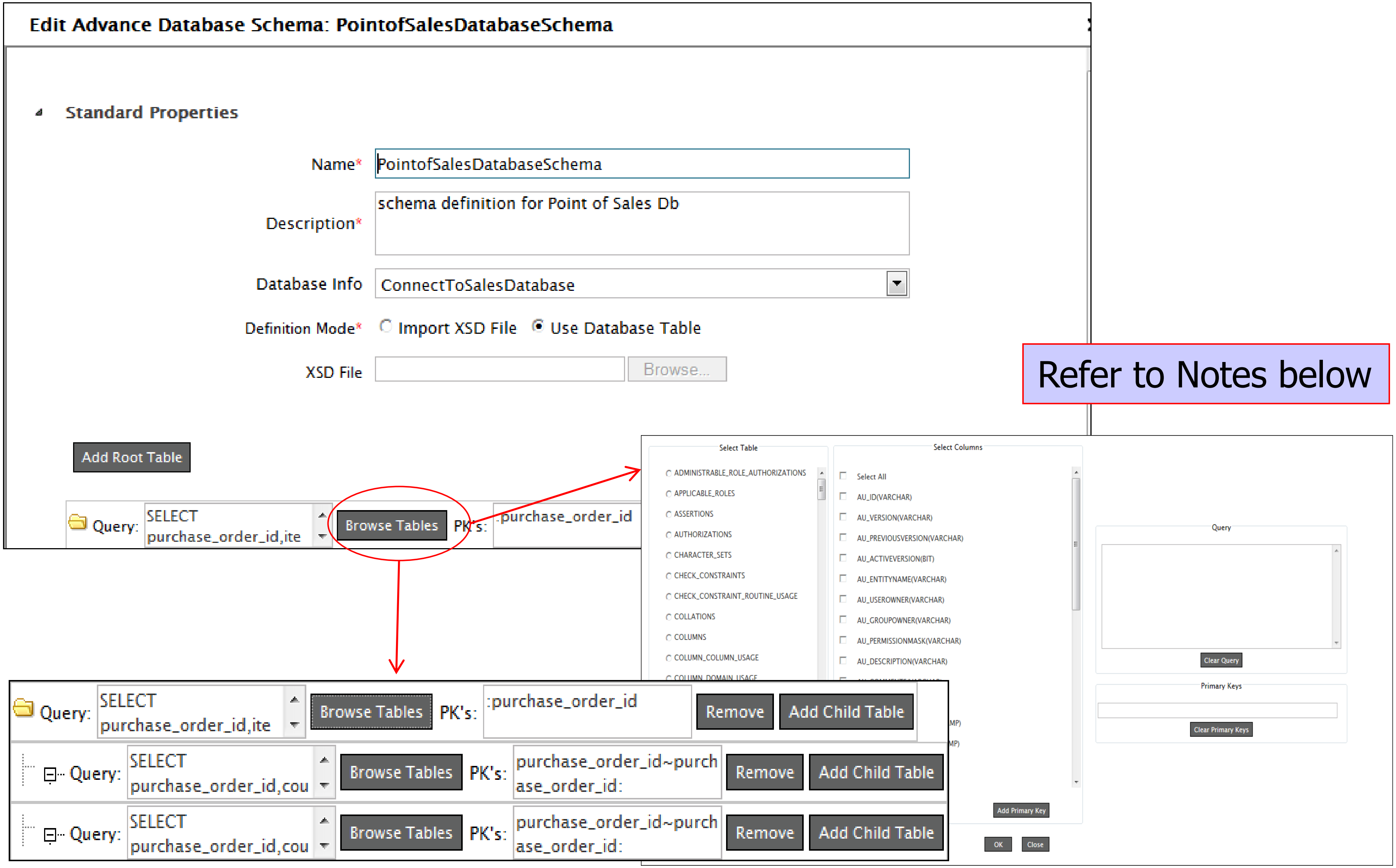
Provider: Not Available

WsProvider XSD Type:

Web Form:

Advanced Properties

Exercise: Purchase Order Processing (Create Database Schema)



Edit Advance Database Schema: PointofSalesDatabaseSchema

Standard Properties

Name* PointofSalesDatabaseSchema

Description* schema definition for Point of Sales Db

Database Info ConnectToSalesDatabase

Definition Mode* ○ Import XSD File ● Use Database Table

XSD File

Query: SELECT purchase_order_id,ite PK's: :purchase_order_id

Select Table

- ○ ADMINISTRABLE_ROLE_AUTHORIZATIONS
- ○ APPLICABLE_ROLES
- ○ ASSERTIONS
- ○ AUTHORIZATIONS
- ○ CHARACTER_SETS
- ○ CHECK_CONSTRAINTS
- ○ CHECK_CONSTRAINT_ROUTINE_USAGE
- ○ COLLATIONS
- ○ COLUMNS
- ○ COLUMN_COLUMN_USAGE
- ○ COLUMN_DOMAIN_USAGE

Select Columns

- □ Select All
- □ AU_ID(VARCHAR)
- □ AU_VERSION(VARCHAR)
- □ AU_PREVIOUSVERSION(VARCHAR)
- □ AU_ACTIVEVERSION(BIT)
- □ AU_ENTITYNAME(VARCHAR)
- □ AU_USEROWNER(VARCHAR)
- □ AU_GROUPOWNER(VARCHAR)
- □ AU_PERMISSIONMASK(VARCHAR)
- □ AU_DESCRIPTION(VARCHAR)

Query

Primary Keys

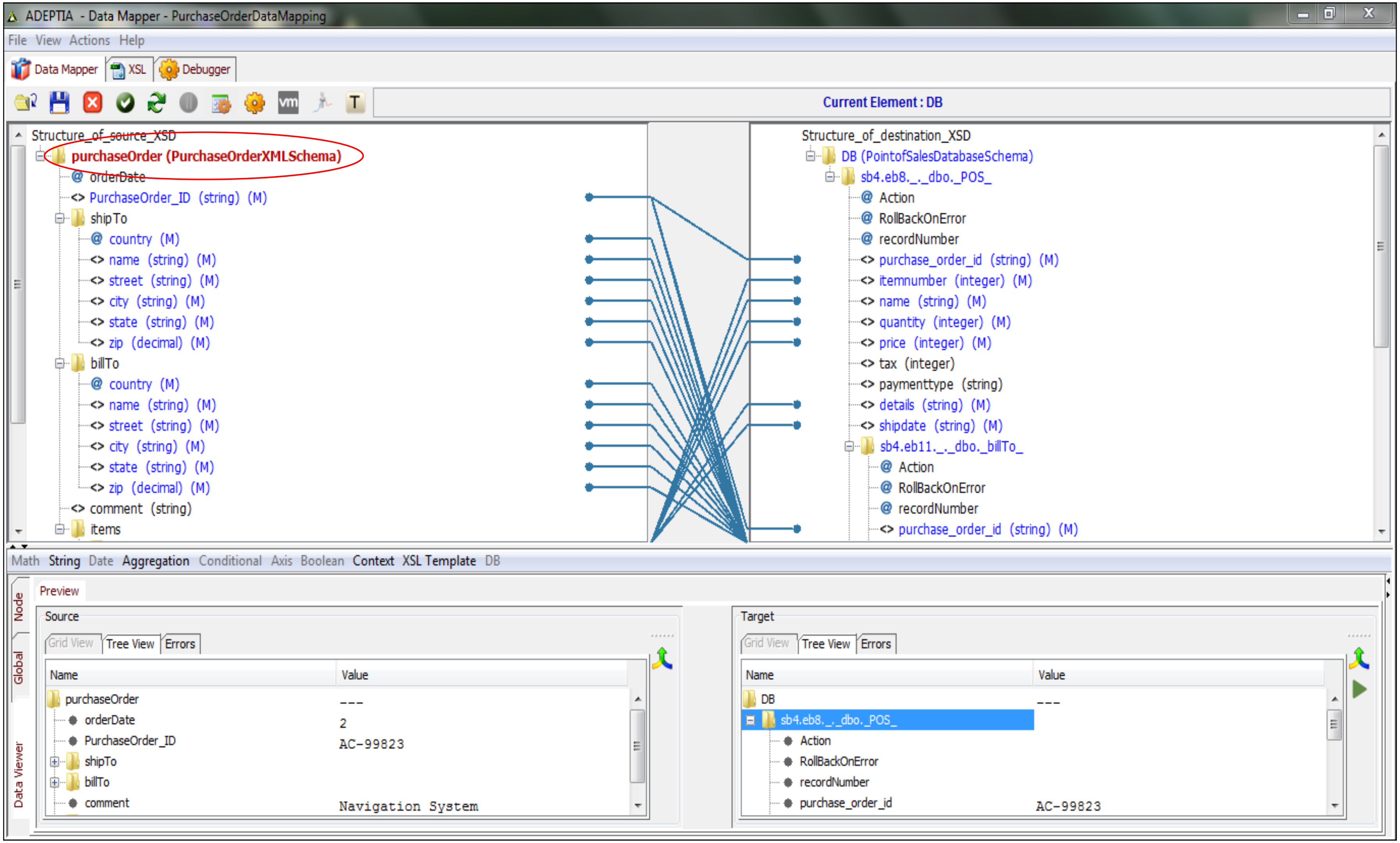
Query: SELECT purchase_order_id,ite PK's: :purchase_order_id

Query: SELECT purchase_order_id,cou PK's: purchase_order_id~purchase_order_id:

Query: SELECT purchase_order_id,cou PK's: purchase_order_id~purchase_order_id:

Refer to Notes below

Exercise: Purchase Order Processing (Create Mapping)



Source

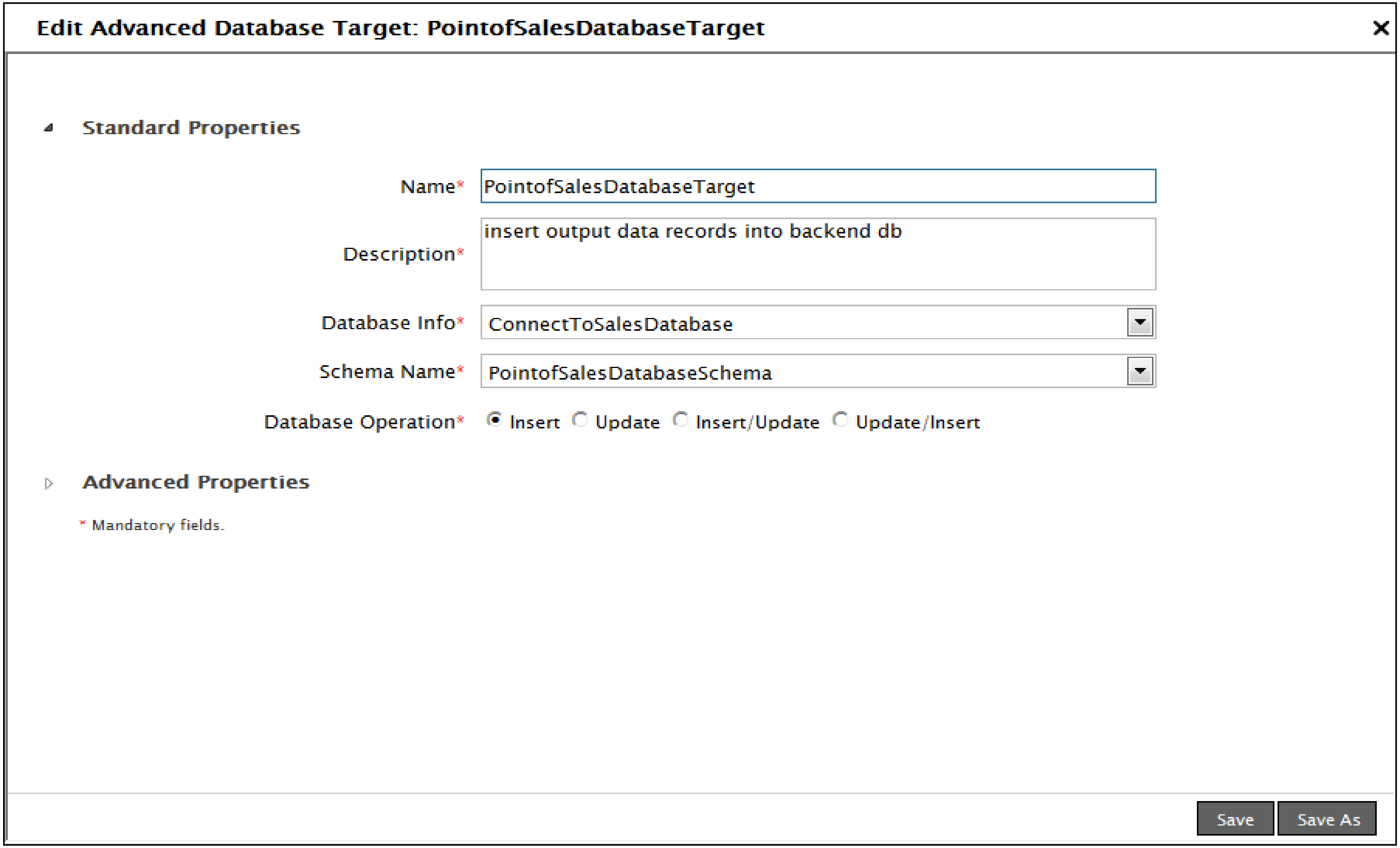
Name	Value
purchaseOrder	---
orderDate	2
PurchaseOrder_ID	AC-99823
shipTo	
billTo	
comment	Navigation System

Target

Name	Value
DB	---
sb4.eb8._._dbo._POS_	
Action	
RollBackOnError	
recordNumber	
purchase_order_id	AC-99823

Refer to Notes below

Exercise: Purchase Order Processing (Create Adv DB Target)



Edit Advanced Database Target: PointofSalesDatabaseTarget

Standard Properties

Name*: PointofSalesDatabaseTarget

Description*: insert output data records into backend db

Database Info*: ConnectToSalesDatabase

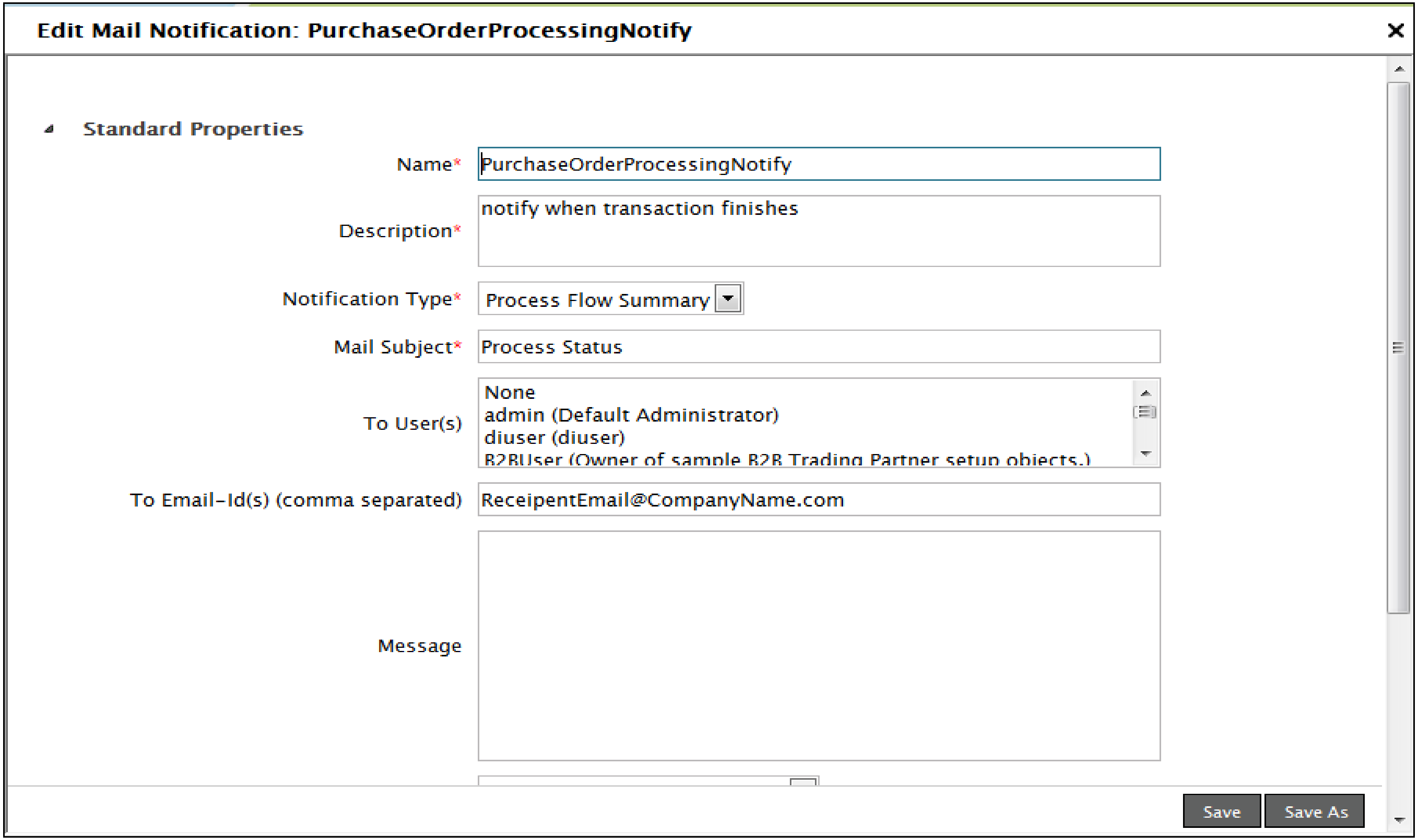
Schema Name*: PointofSalesDatabaseSchema

Database Operation*: ● Insert ○ Update ○ Insert/Update ○ Update/Insert

Advanced Properties

* Mandatory fields.

Exercise: Purchase Order Processing (Create Notification)

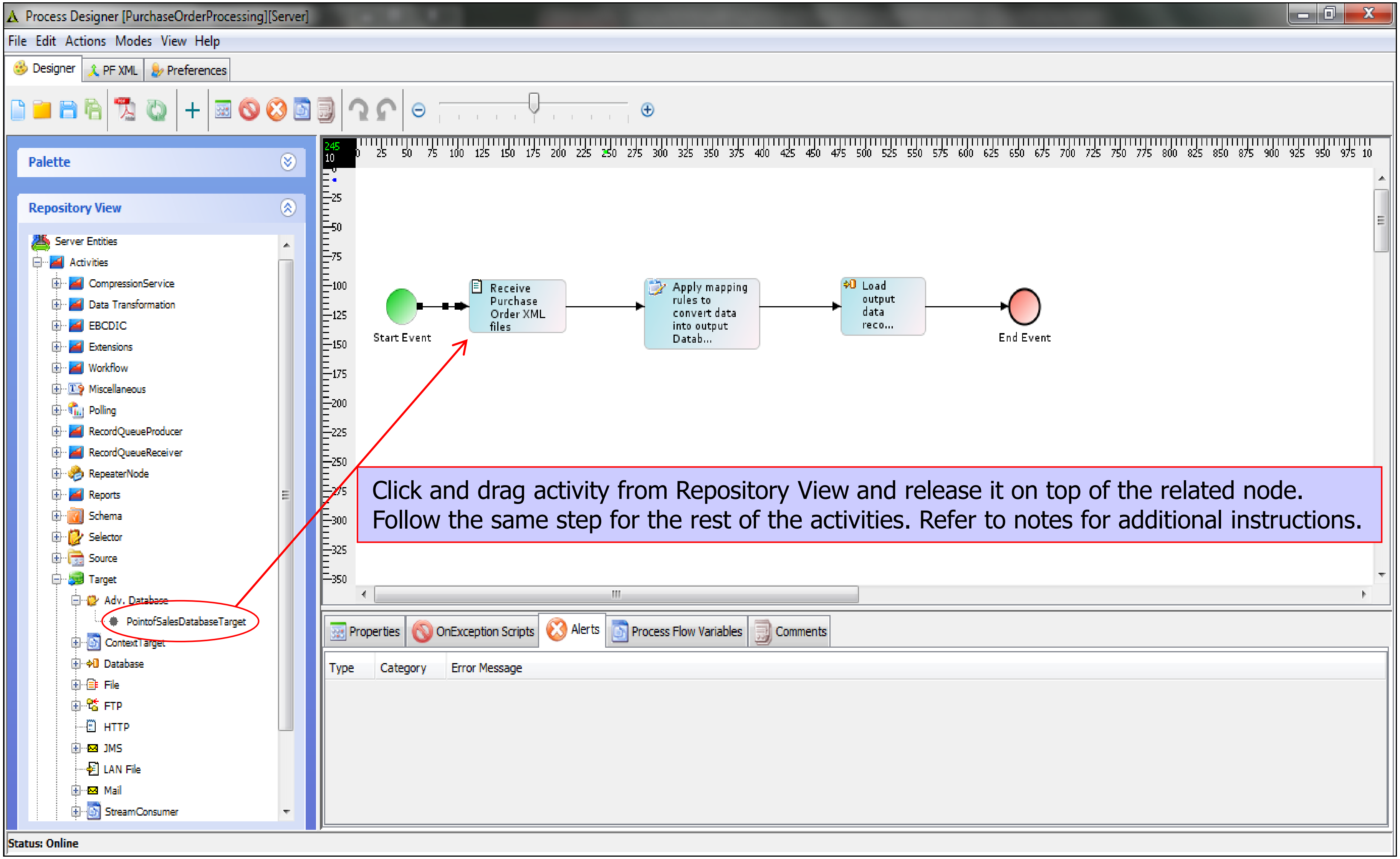


Edit Mail Notification: PurchaseOrderProcessingNotify

Standard Properties

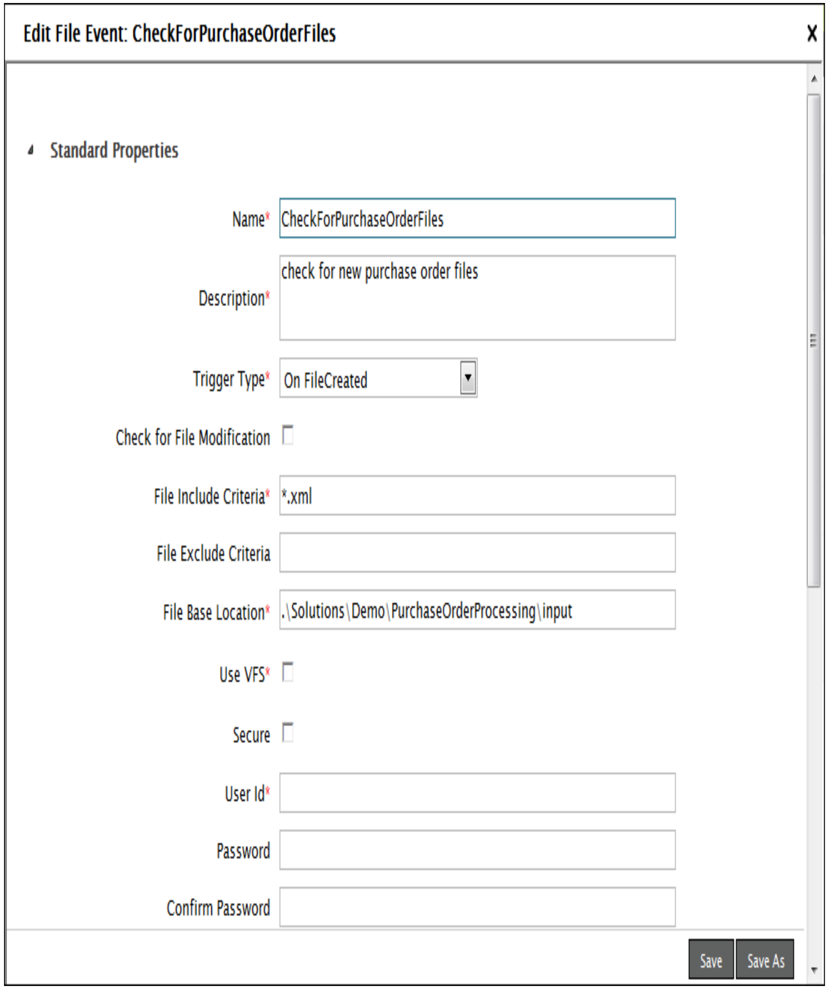
Name*	PurchaseOrderProcessingNotify
Description*	notify when transaction finishes
Notification Type*	Process Flow Summary
Mail Subject*	Process Status
To User(s)	None admin (Default Administrator) diuser (diuser) B2BUser (Owner of sample B2B Trading Partner setup objects.)
To Email-Id(s) (comma separated)	ReceipentEmail@CompanyName.com
Message	

Exercise: Purchase Order Processing (configure process flow)



Click and drag activity from Repository View and release it on top of the related node. Follow the same step for the rest of the activities. Refer to notes for additional instructions.

Exercise: Purchase Order Processing (deploy process flow)



Edit File Event: CheckForPurchaseOrderFiles

Standard Properties

Name*: CheckForPurchaseOrderFiles

Description*: check for new purchase order files

Trigger Type*: On FileCreated

Check for File Modification □

File Include Criteria*: *.xml

File Exclude Criteria:

File Base Location*: .\Solutions\Demo\PurchaseOrderProcessing\input

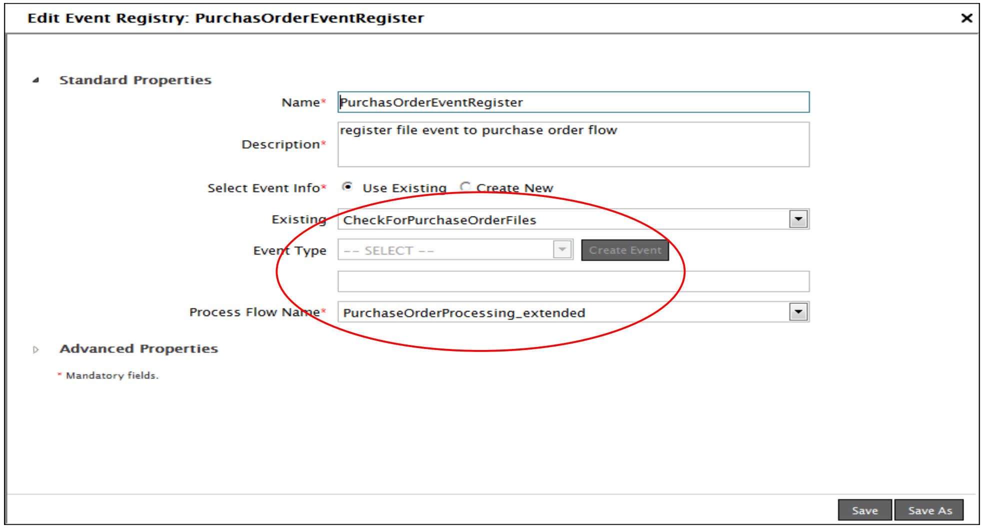
Use VFS* □

Secure □

User Id*:

Password:

Confirm Password:



Edit Event Registry: PurchasOrderEventRegister

Standard Properties

Name*: PurchasOrderEventRegister

Description*: register file event to purchase order flow

Select Event Info*: ● Use Existing ○ Create New

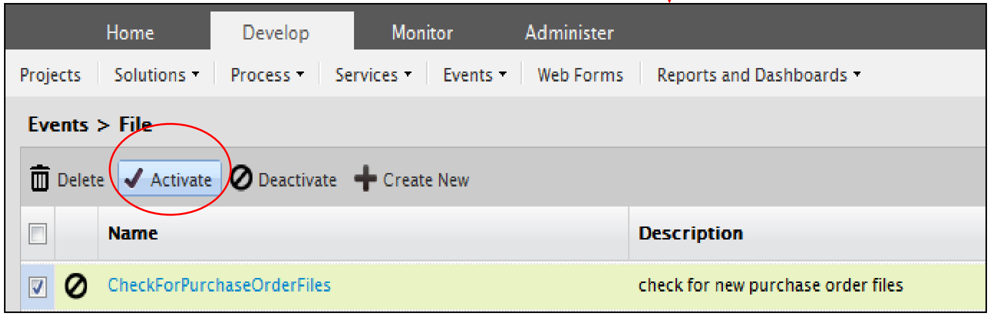
Existing: CheckForPurchaseOrderFiles

Event Type: -- SELECT --

Process Flow Name*: PurchaseOrderProcessing_extended

Advanced Properties

* Mandatory fields.



□		Name	Description
☑	⊘	CheckForPurchaseOrderFiles	check for new purchase order files

Refer to Notes below

Exercise: Purchase Order Processing Extended

Exercise: Purchase Order Processing _Extended

This process flow is an extension of the previous exercise. We will modify the design of the existing Purchase Order process and add a rule. First edit the existing PurchaseOrderProcessing flow, rename it to PurchaseOrderProcessing_extended and click on SAVE AS to make a copy.

Design:

- Apply a rule that will check for the Item Quantity in the source file and if it is greater than 100, process will not load the data into Database and instead it will email the Source file to a User as an Email attachment. If the Quantity is less than 100 then the process will load the data into the database.

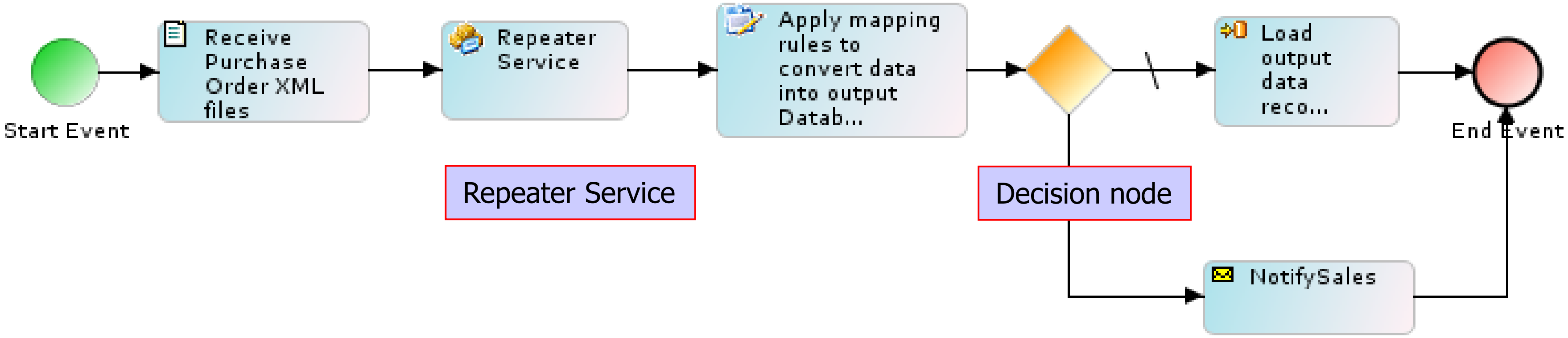
Configure:

- Configure the process by creating below activities and attaching them to the appropriate steps in the process flow:
  - Create Email Target activity
  - Modify existing PurchaseOrderMapping activity
  - Modify the Process flow to include the Quantity rule

Deploy: <the below steps are already completed in the previous exercise and should not be repeated>

- Create a File Event
- Register the File Event with Purchase Order Processing process flow
- Activate File Event

Exercise: Purchase Order Processing Extended


Start Event
Receive Purchase Order XML files
Repeater Service
Apply mapping rules to convert data into output Datab...
Load output data reco...
End Event
NotifySales

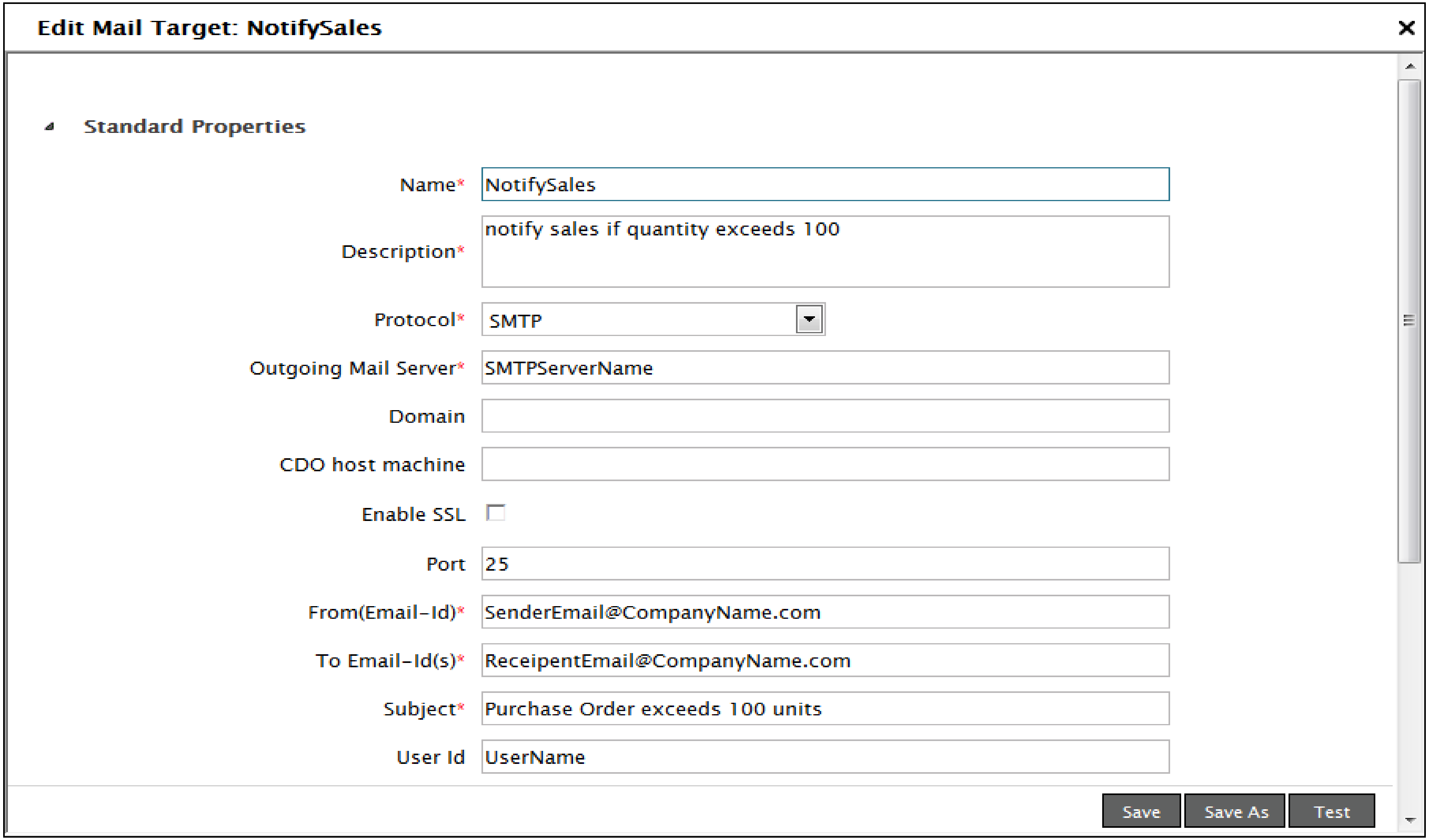
Repeater Service

Decision node

Email target

Refer to Notes below

Exercise: Purchase Order Processing (Create Email Target)



Edit Mail Target: NotifySales

Standard Properties

Name* NotifySales

Description* notify sales if quantity exceeds 100

Protocol* SMTP

Outgoing Mail Server* SMTPServerName

Domain

CDO host machine

Enable SSL □

Port 25

From(Email-Id)* SenderEmail@CompanyName.com

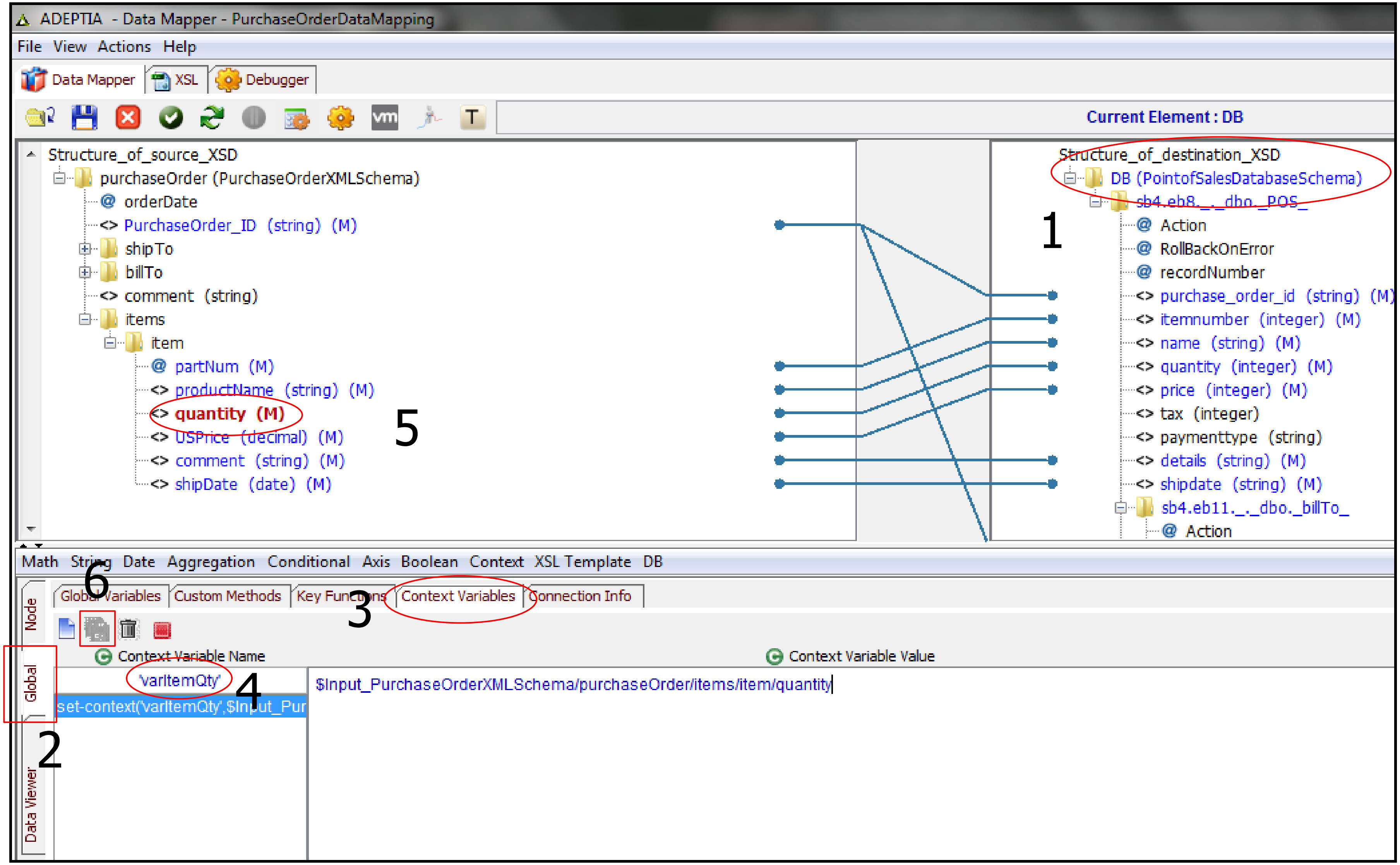
To Email-Id(s)* ReceipentEmail@CompanyName.com

Subject* Purchase Order exceeds 100 units

User Id UserName

Refer to Notes below

Exercise: Purchase Order Processing (Edit Mapping)



ADEPTIA - Data Mapper - PurchaseOrderDataMapping

File View Actions Help

Data Mapper | XSL | Debugger

Current Element : DB

```
Structure_of_source_XSD
  purchaseOrder (PurchaseOrderXMLSchema)
    @ orderDate
    <> PurchaseOrder_ID (string) (M)
    shipTo
    billTo
    <> comment (string)
    items
      item
        @ partNum (M)
        <> productName (string) (M)
        <> quantity (M)
        <> USPrice (decimal) (M)
        <> comment (string) (M)
        <> shipDate (date) (M)
```

```
Structure_of_destination_XSD
  DB (PointofSalesDatabaseSchema)
    sb4.eb8._._dbo._POS_
      @ Action
      @ RollBackOnError
      @ recordNumber
      <> purchase_order_id (string) (M)
      <> itemnumber (integer) (M)
      <> name (string) (M)
      <> quantity (integer) (M)
      <> price (integer) (M)
      <> tax (integer)
      <> paymenttype (string)
      <> details (string) (M)
      <> shipdate (string) (M)
    sb4.eb11._._dbo._billTo_
      @ Action
```

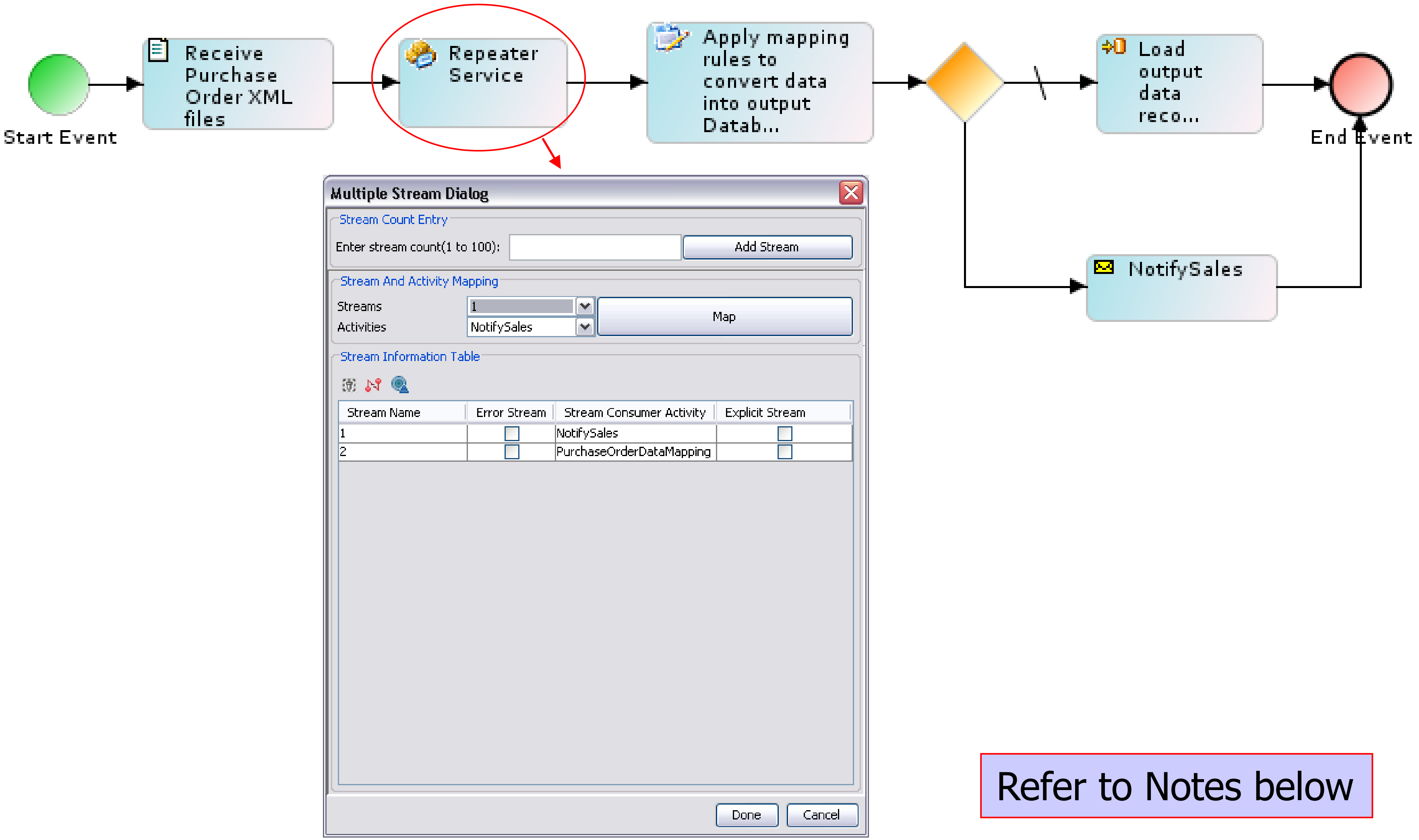
Math String Date Aggregation Conditional Axis Boolean Context XSL Template DB

Global Variables | Custom Methods | Key Functions | Context Variables | Connection Info

Context Variable Name	Context Variable Value
'varItemQty'	$Input_PurchaseOrderXMLSchema/purchaseOrder/items/item/quantity
set-context('varItemQty',$Input_Pur	

Refer to Notes below

Exercise: Purchase Order Processing Extended (Edit Process Flow)


Start Event
Receive Purchase Order XML files
Repeater Service
Apply mapping rules to convert data into output Datab...
Load output data reco...
End Event
NotifySales

Multiple Stream Dialog

Stream Count Entry

Enter stream count(1 to 100):

Stream And Activity Mapping

Streams: 1

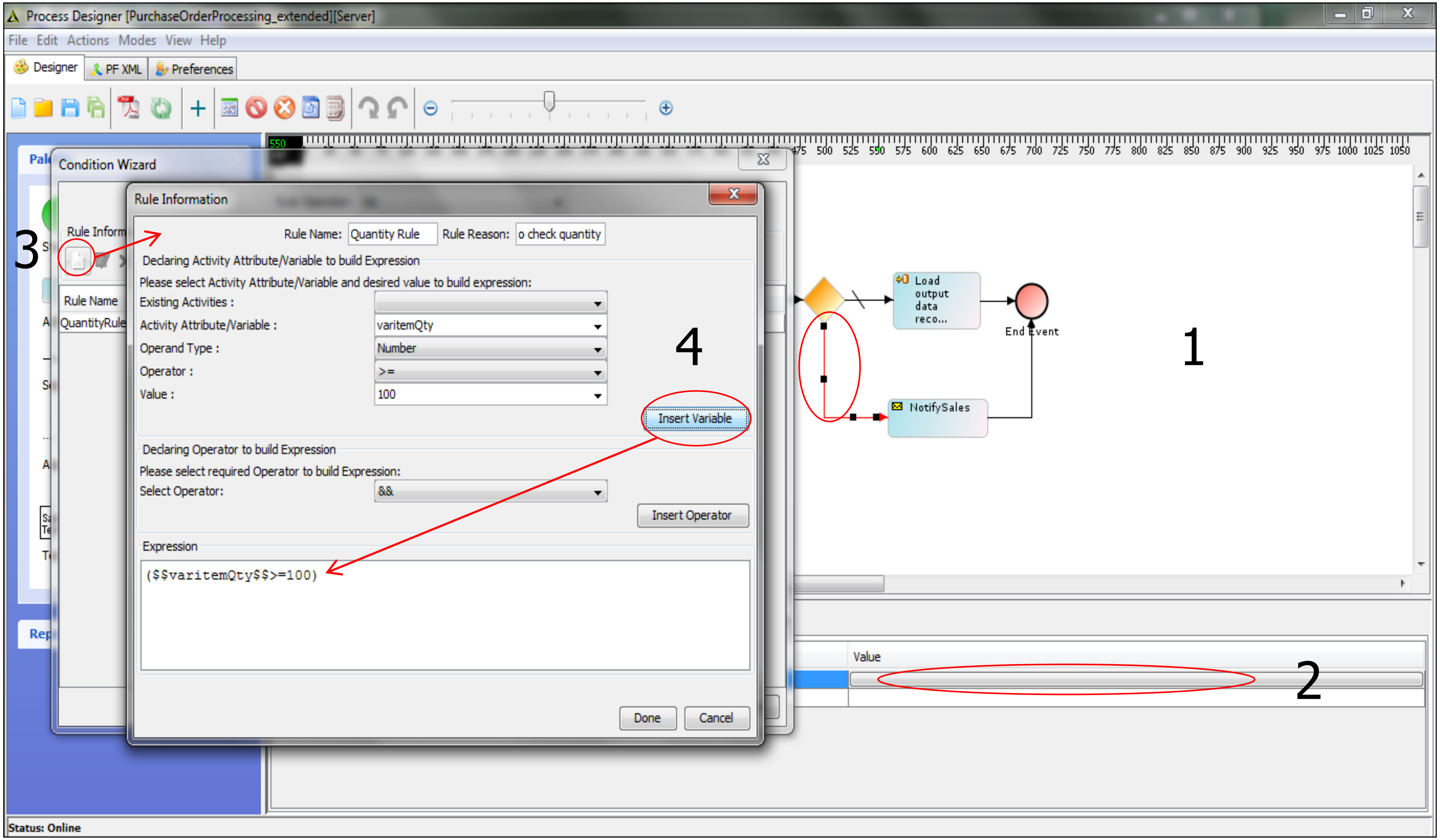
Activities: NotifySales

Stream Information Table

Stream Name	Error Stream	Stream Consumer Activity	Explicit Stream
1	□	NotifySales	□
2	□	PurchaseOrderDataMapping	□

Refer to Notes below

Exercise: Purchase Order Processing Extended (Edit Process Flow)



Rule Name: Quantity Rule Rule Reason: o check quantity

Declaring Activity Attribute/Variable to build Expression

Please select Activity Attribute/Variable and desired value to build expression:

- Existing Activities :
- Activity Attribute/Variable : varitemQty
- Operand Type : Number
- Operator : >=
- Value : 100

Insert Variable

Declaring Operator to build Expression

Please select required Operator to build Expression:

Select Operator: &&

Insert Operator

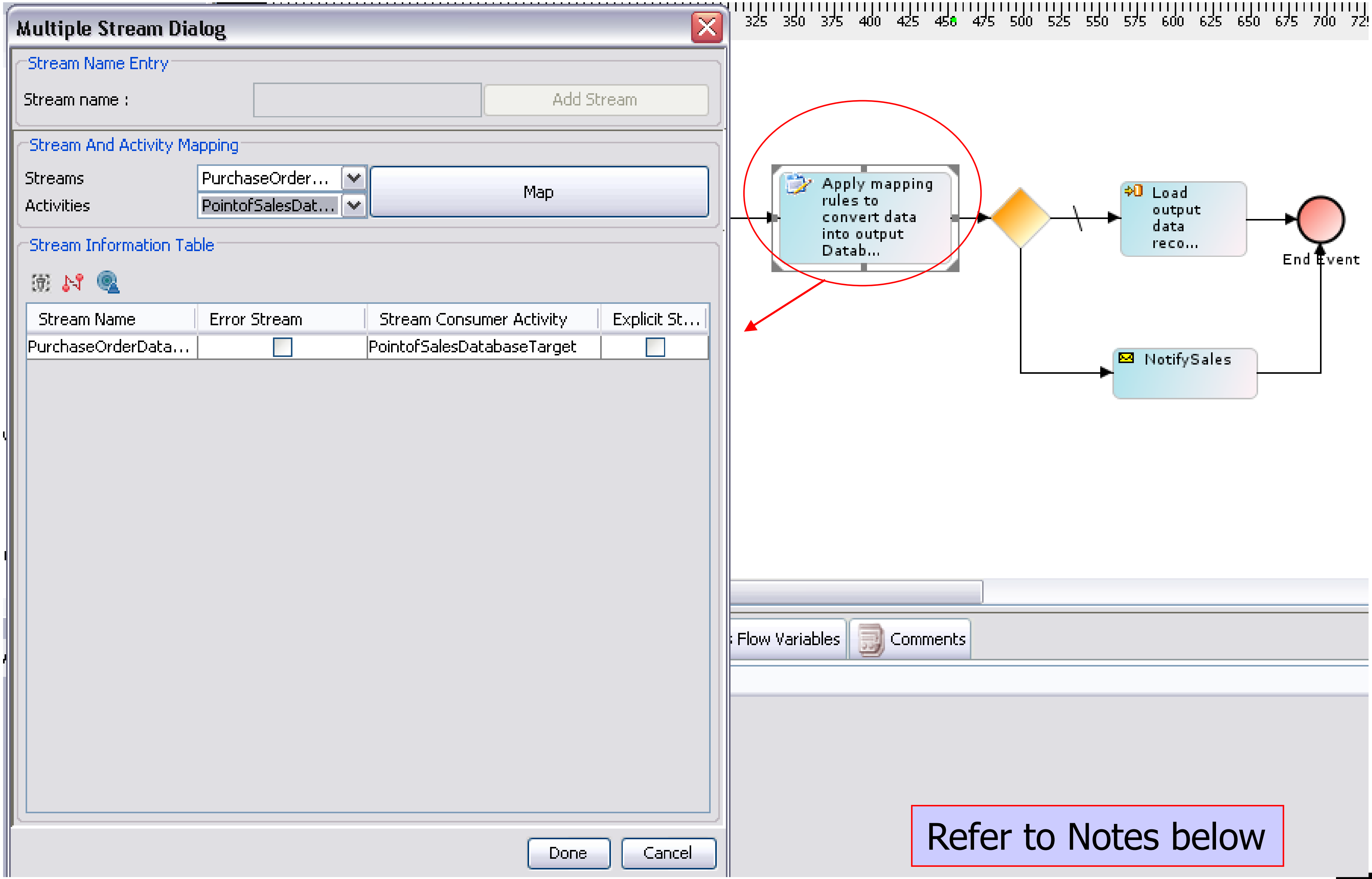
Expression

```
($$varitemQty$$>=100)
```

Done Cancel

Refer to Notes below

Exercise: Purchase Order Processing Extended (Edit Process Flow)



Multiple Stream Dialog

Stream Name Entry

Stream name : [Add Stream]

Stream And Activity Mapping

Streams: PurchaseOrder...

Activities: PointofSalesDat...

[Map]

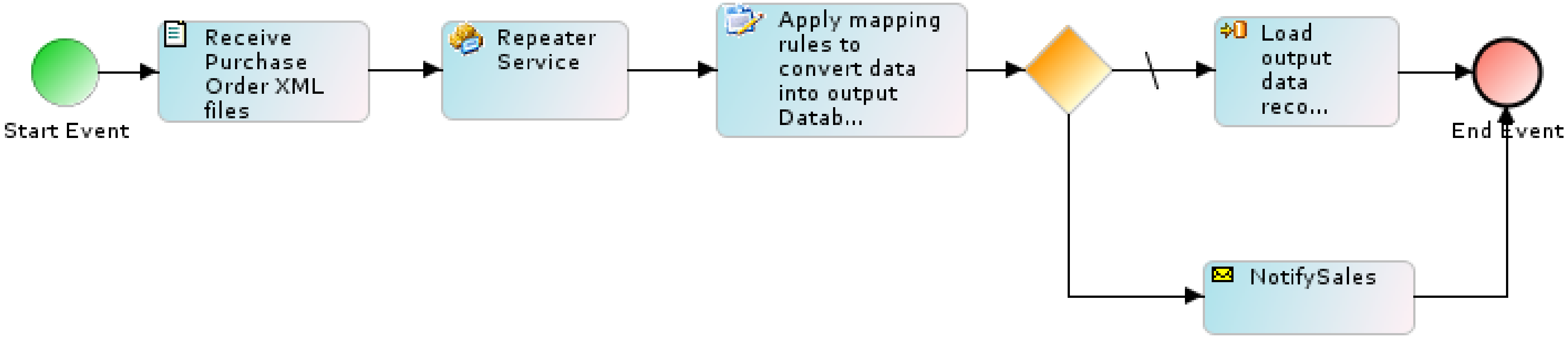
Stream Information Table

Stream Name	Error Stream	Stream Consumer Activity	Explicit St...
PurchaseOrderData...	□	PointofSalesDatabaseTarget	□

[Done] [Cancel]

Refer to Notes below

Exercise: Purchase Order Processing Extended (final)


Start Event
Receive Purchase Order XML files
Repeater Service
Apply mapping rules to convert data into output Datab...
Load output data reco...
End Event
NotifySales

Refer to Notes below

Exercise: Order Fulfillment Process

Exercise: Order Fulfillment Process

Order Fulfillment process automatically gets triggered when a Shipping department receives an email with customer orders from Sales. After receiving the email this process calls on a Fulfillment Web Service to fulfill these orders. Fulfillment Web Service responds back with Shipping and Invoice information for each order. Order Fulfillment process updates this information back into the customer order data file and emails it back to Sales.

Design:

- Receive Email containing customer orders from Sales
- Create Web Service Request message
- Call on a Fulfillment Web Service (prerequisite: create a sub process and publish it as a web service)
- Map Web Service Response back to customer orders
- Email the updated customer orders (shipping and invoice columns populated) back to Sales
- If Web Service does not respond send an error notification via email to Administrator

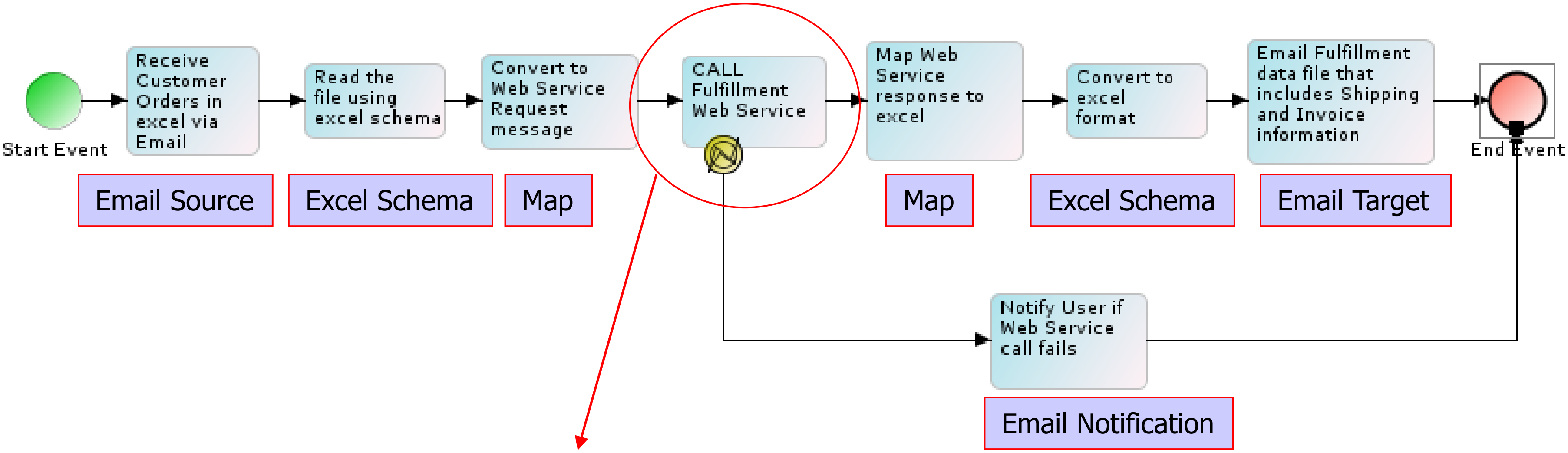
Configure:

- Configure the process by creating below activities
  - Create Email Source and Target activity
  - Create a sub-process and publish it as a web service
  - Create mappings for generating SOAP Request and Response
  - Create Web Service Provider and Consumer activities

Deploy:

- Create a Email Event
- Register the Email Event with Order Fulfillment process flow
- Activate Email Event

Exercise: Order Fulfillment Process (design)


Start Event
Receive Customer Orders in excel via Email
Read the file using excel schema
Convert to Web Service Request message
CALL Fulfillment Web Service
Map Web Service response to excel
Convert to excel format
Email Fulfillment data file that includes Shipping and Invoice information
End Event
Notify User if Web Service call fails

Email Source

Excel Schema

Map

Map

Excel Schema

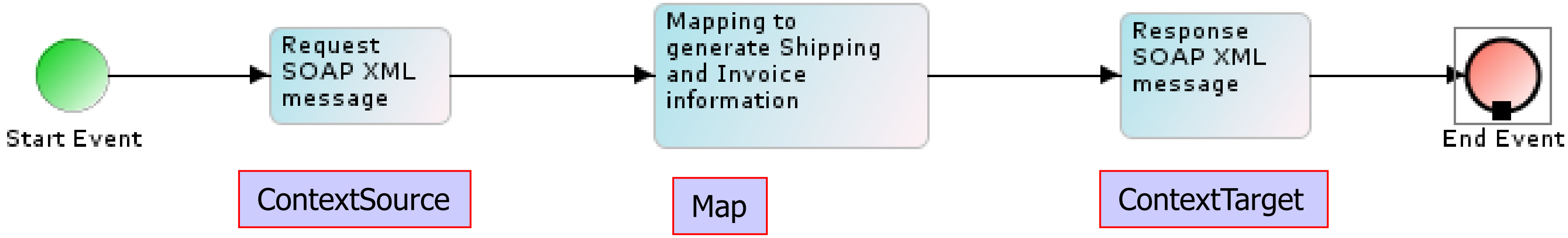
Email Target

Email Notification

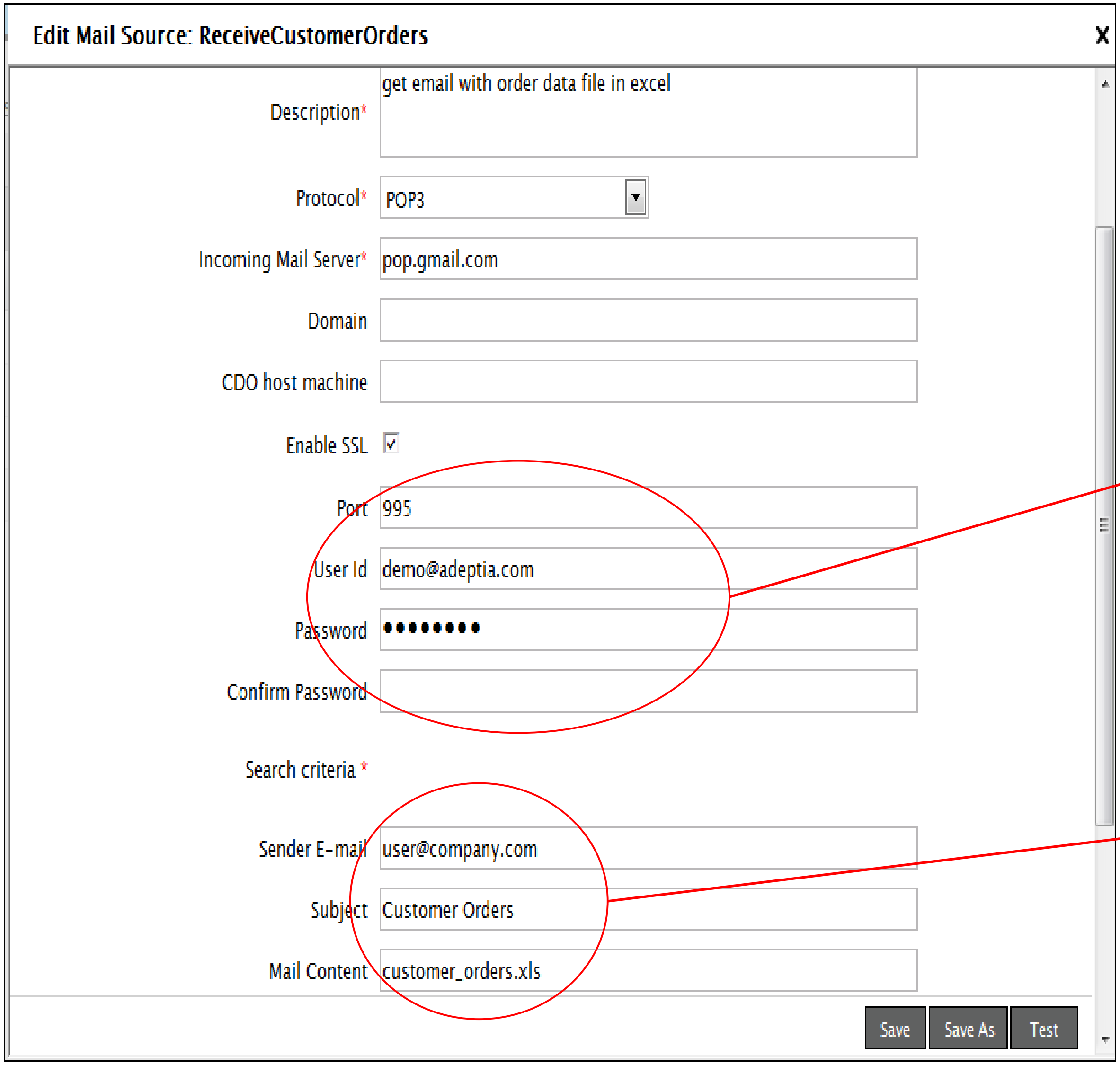
Web Service Message call
(Web Service consumer).
This step calls on a sub-process.

Refer to Notes below

Exercise: Fulfillment WebService subprocess (design)


Start Event
Request SOAP XML message
ContextSource
Mapping to generate Shipping and Invoice information
Map
Response SOAP XML message
ContextTarget
End Event

Exercise: Order Fulfillment Process (Create Email Source)



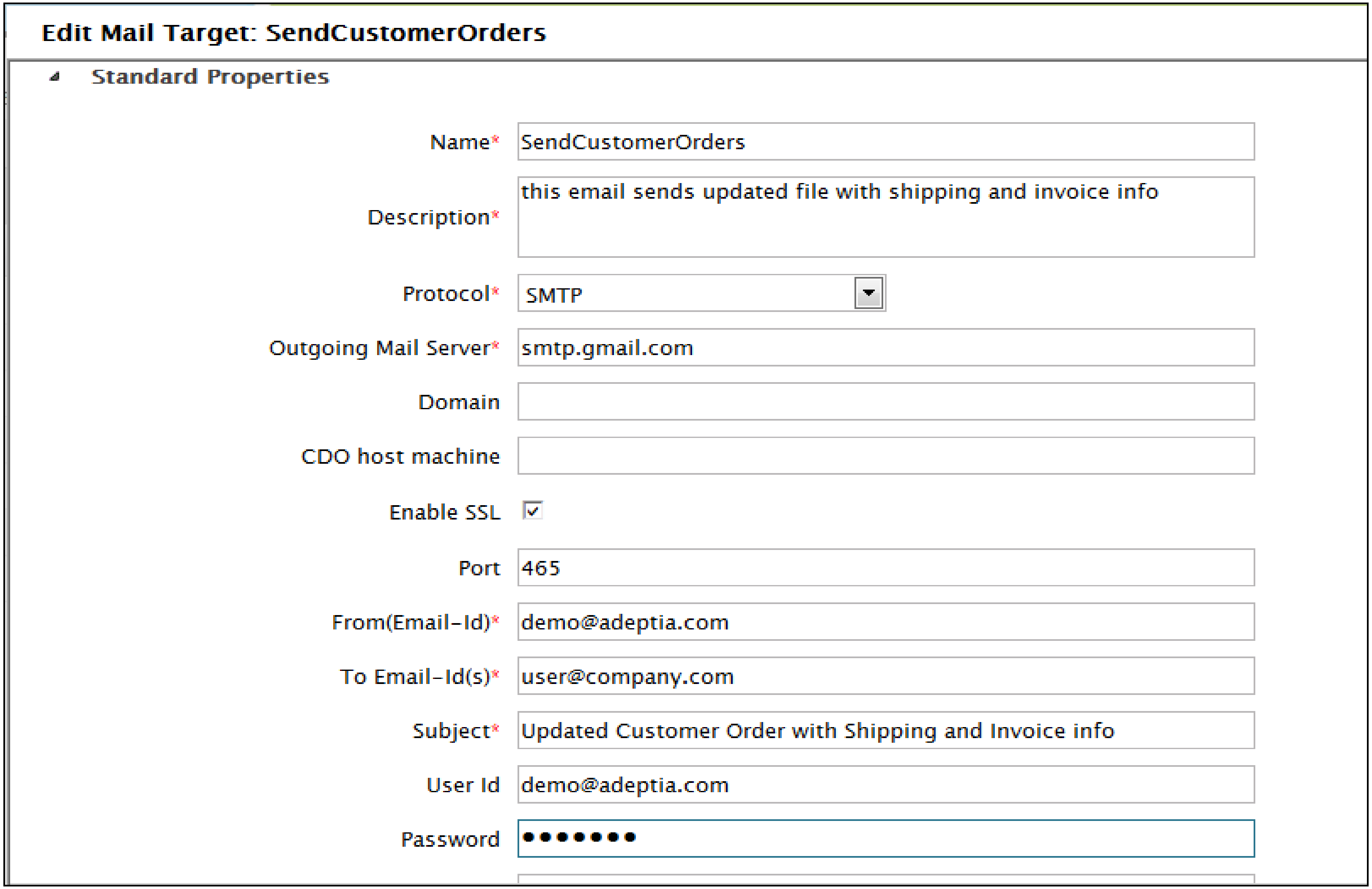
Edit Mail Source: ReceiveCustomerOrders

Description*	get email with order data file in excel
Protocol*	POP3
Incoming Mail Server*	pop.gmail.com
Domain	
CDO host machine	
Enable SSL	☑
Port	995
User Id	demo@adeptia.com
Password	••••••••
Confirm Password	
Search criteria *	
Sender E-mail	user@company.com
Subject	Customer Orders
Mail Content	customer_orders.xls

Recipient Email address that will receive Customer order email. The login account is for Adeptia application to login and look for that Email.

Adeptia application searches for the email based on this criteria.

Exercise: Order Fulfillment Process (Create Email Target)

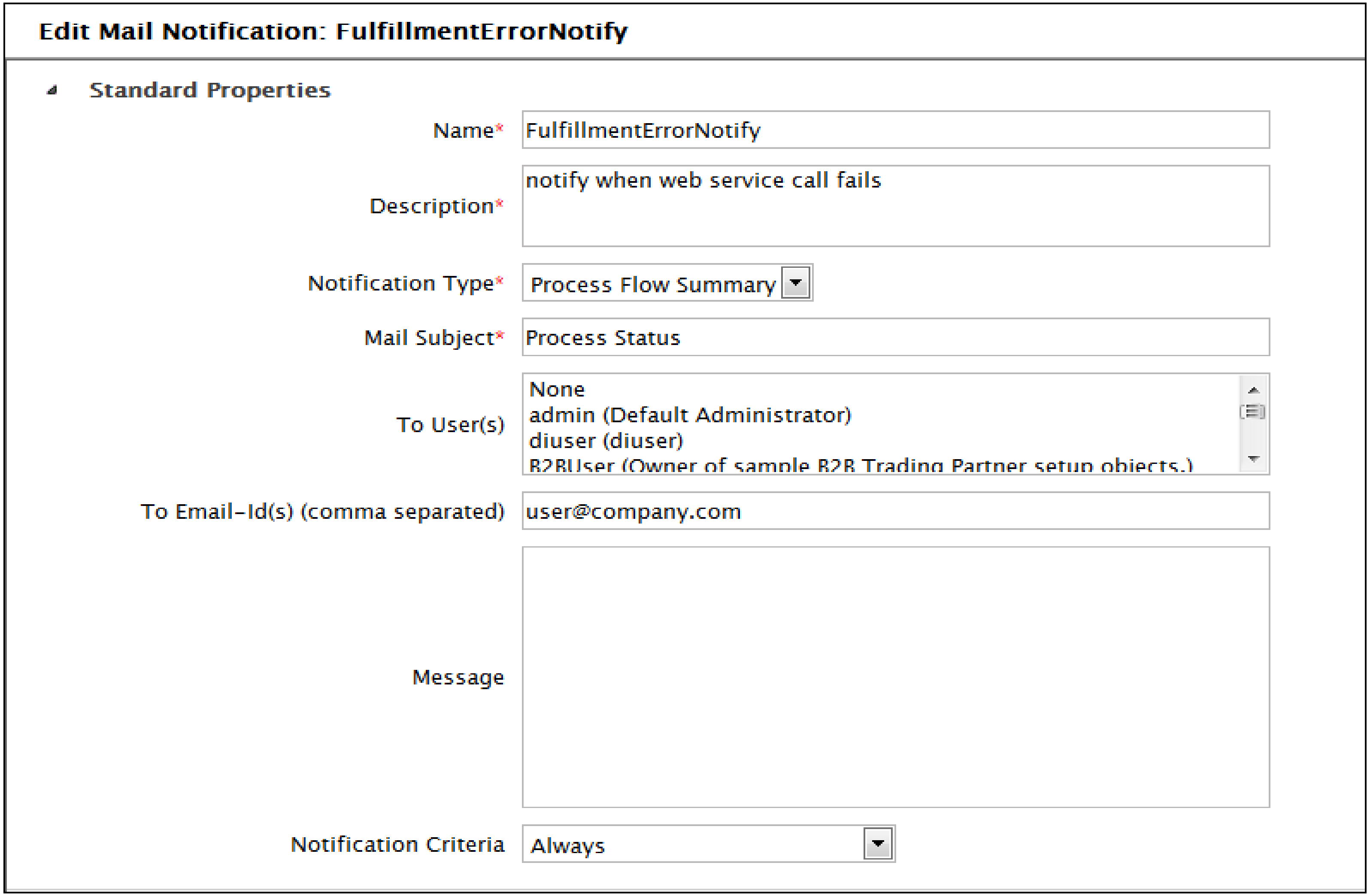


Edit Mail Target: SendCustomerOrders

Standard Properties

Name*	SendCustomerOrders
Description*	this email sends updated file with shipping and invoice info
Protocol*	SMTP
Outgoing Mail Server*	smtp.gmail.com
Domain	
CDO host machine	
Enable SSL	☑
Port	465
From(Email-Id)*	demo@adeptia.com
To Email-Id(s)*	user@company.com
Subject*	Updated Customer Order with Shipping and Invoice info
User Id	demo@adeptia.com
Password	••••••••

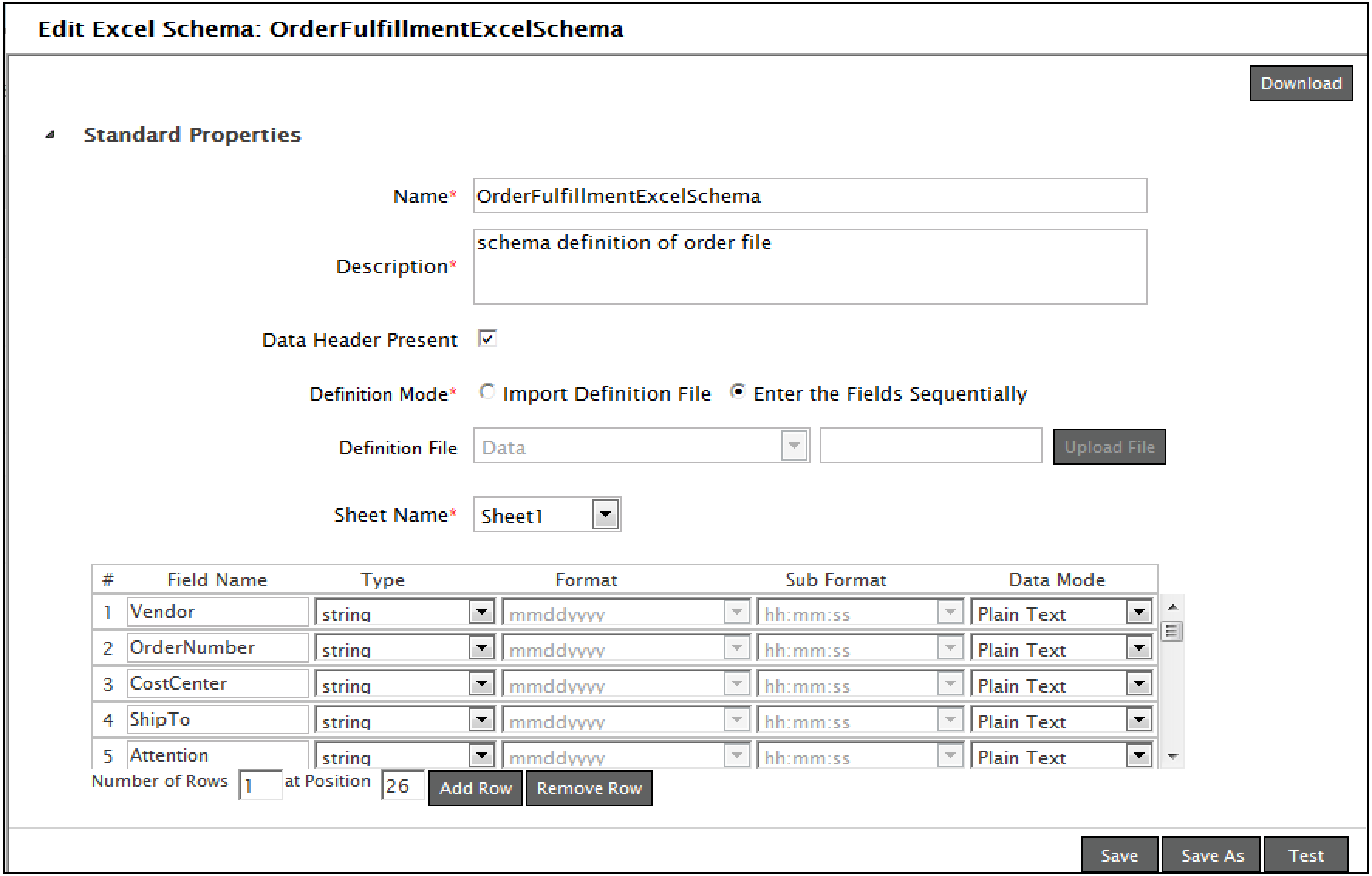
Exercise: Order Fulfillment Process (Create Email Notification)



Edit Mail Notification: FulfillmentErrorNotify

Name*	FulfillmentErrorNotify
Description*	notify when web service call fails
Notification Type*	Process Flow Summary
Mail Subject*	Process Status
To User(s)	None admin (Default Administrator) diuser (diuser) B2BUser (Owner of sample B2B Trading Partner setup objects.)
To Email-Id(s) (comma separated)	user@company.com
Message	
Notification Criteria	Always

Exercise: Order Fulfillment Process (Create Excel Schema)



Edit Excel Schema: OrderFulfillmentExcelSchema

Standard Properties

Name*: OrderFulfillmentExcelSchema

Description*: schema definition of order file

Data Header Present ☑

Definition Mode*: ○ Import Definition File ● Enter the Fields Sequentially

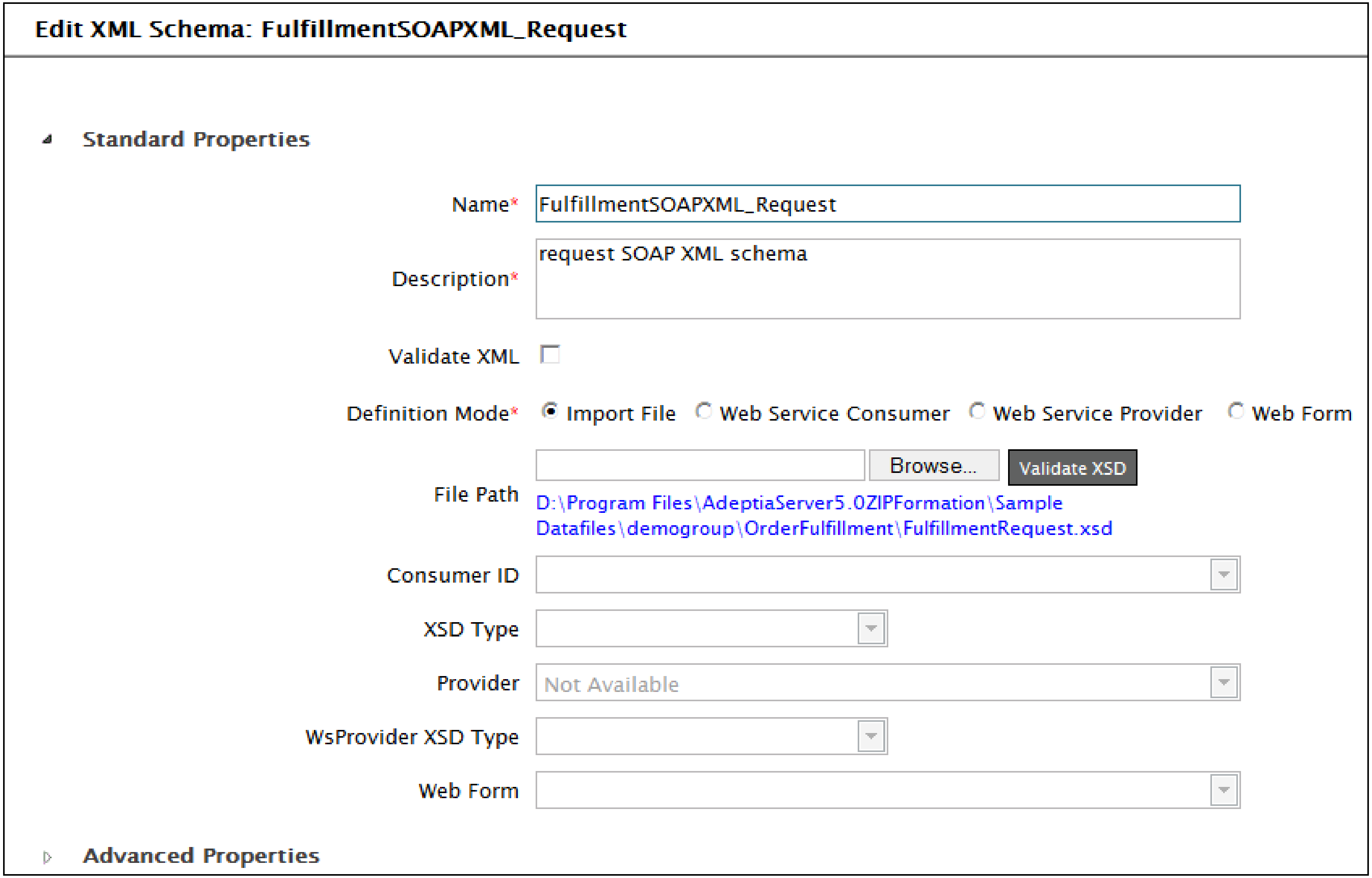
Definition File: Data

Sheet Name*: Sheet1

#	Field Name	Type	Format	Sub Format	Data Mode
1	Vendor	string	mmddyyyy	hh:mm:ss	Plain Text
2	OrderNumber	string	mmddyyyy	hh:mm:ss	Plain Text
3	CostCenter	string	mmddyyyy	hh:mm:ss	Plain Text
4	ShipTo	string	mmddyyyy	hh:mm:ss	Plain Text
5	Attention	string	mmddyyyy	hh:mm:ss	Plain Text

Number of Rows 1 at Position 26

Exercise: Order Fulfillment Process (Create XML request Schema)



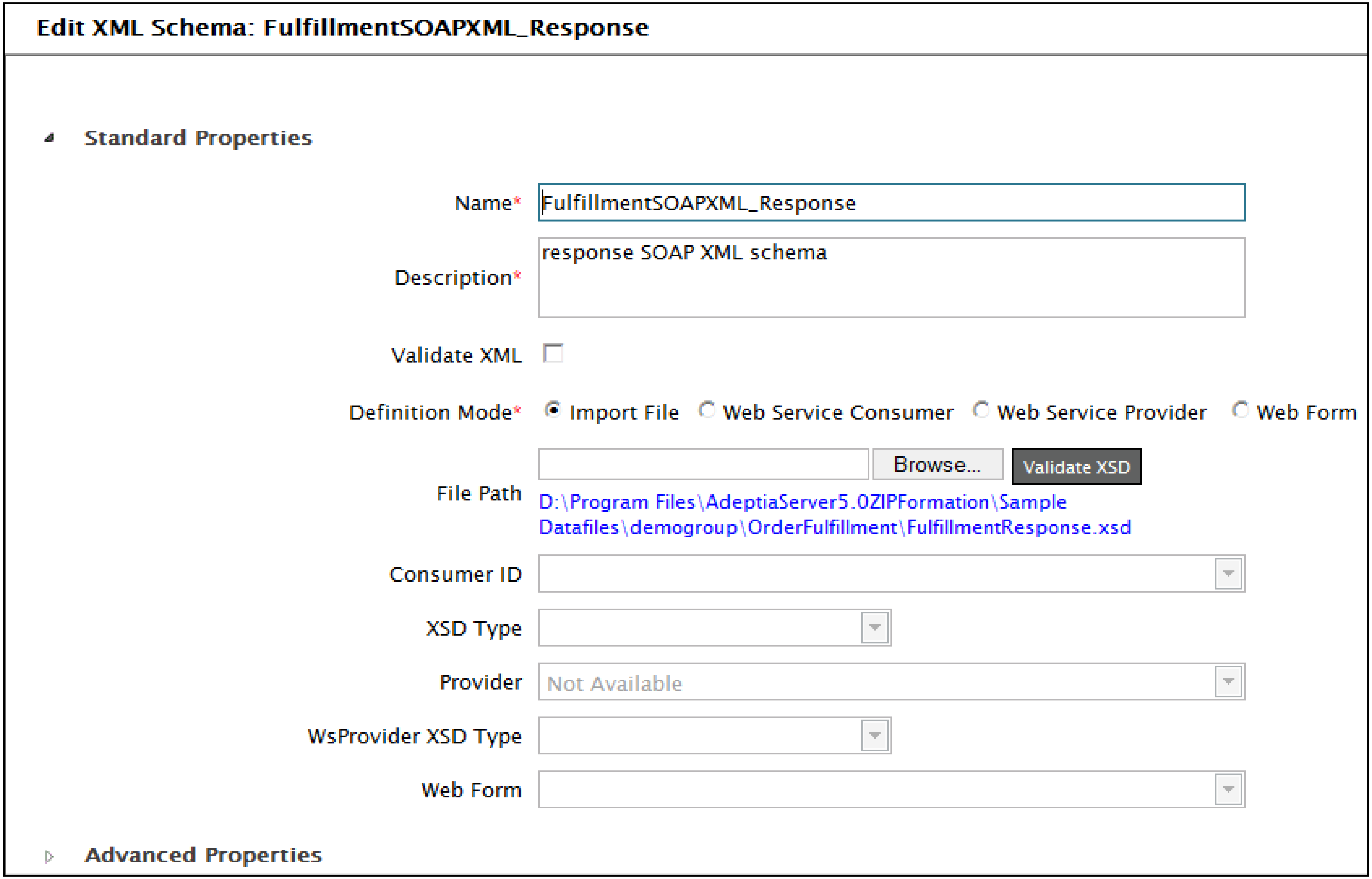
Edit XML Schema: FulfillmentSOAPXML_Request

Standard Properties

Name*	FulfillmentSOAPXML_Request
Description*	request SOAP XML schema
Validate XML	□
Definition Mode*	● Import File ○ Web Service Consumer ○ Web Service Provider ○ Web Form
File Path	D:\Program Files\AdeptiaServer5.0ZIPFormation\Sample Datafiles\demogroup\OrderFulfillment\FulfillmentRequest.xsd
Consumer ID	
XSD Type	
Provider	Not Available
WsProvider XSD Type	
Web Form	

Advanced Properties

Exercise: Order Fulfillment Process (Create XML response Schema)

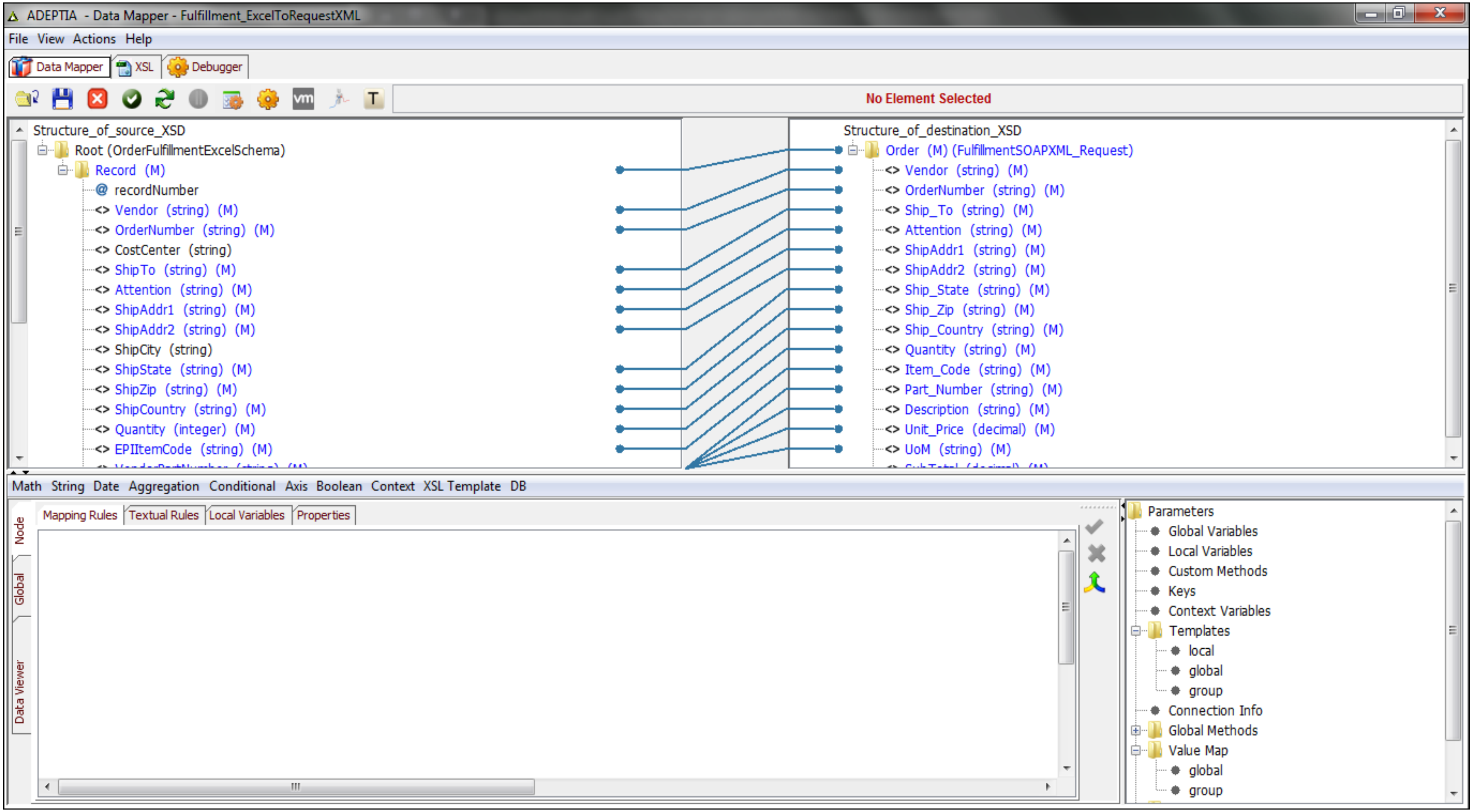


Edit XML Schema: FulfillmentSOAPXML_Response

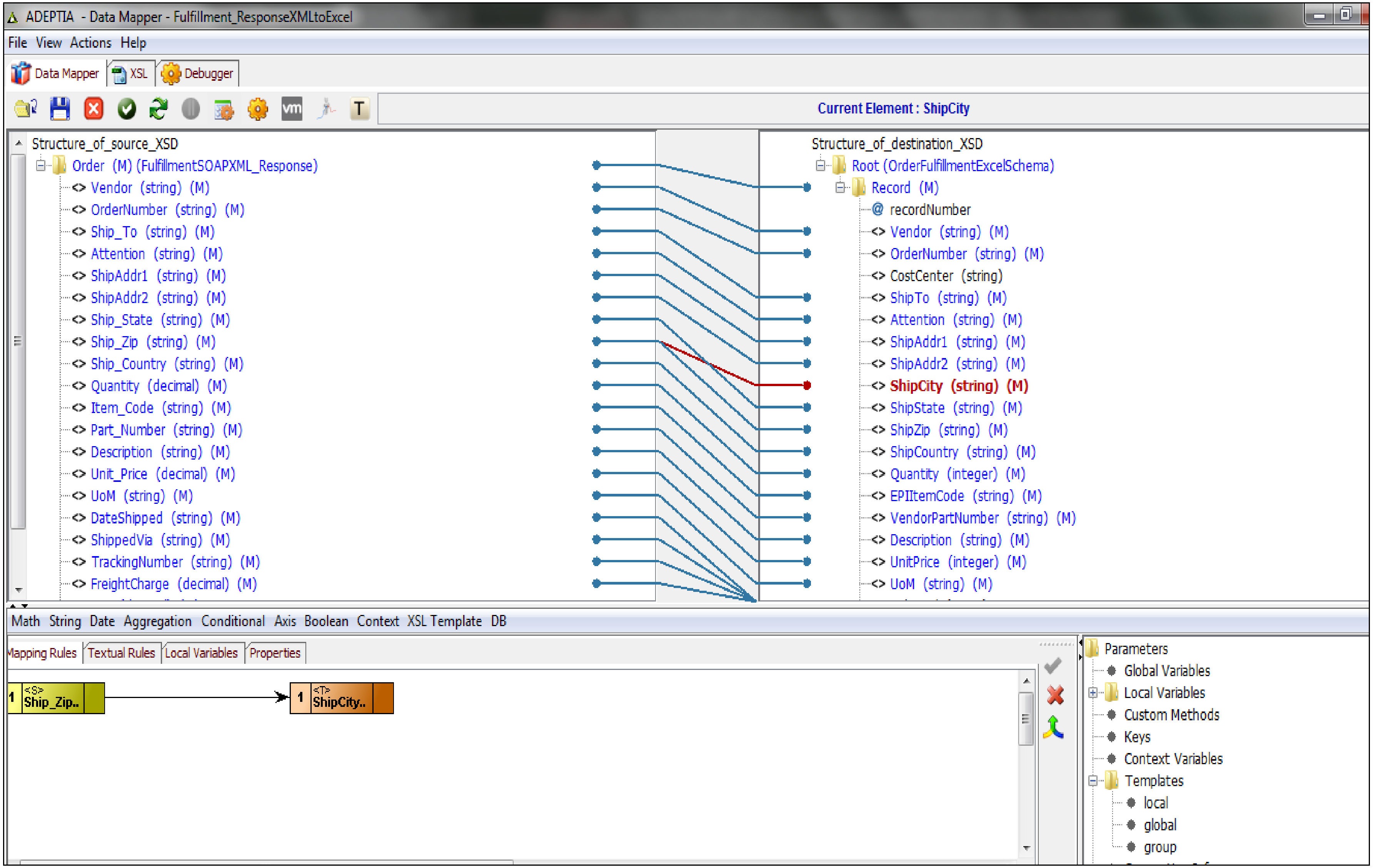
Standard Properties

Name*	FulfillmentSOAPXML_Response
Description*	response SOAP XML schema
Validate XML	□
Definition Mode*	● Import File ○ Web Service Consumer ○ Web Service Provider ○ Web Form
File Path	D:\Program Files\AdeptiaServer5.0ZIPFormation\Sample Datafiles\demogroup\OrderFulfillment\FulfillmentResponse.xsd
Consumer ID	
XSD Type	
Provider	Not Available
WsProvider XSD Type	
Web Form	

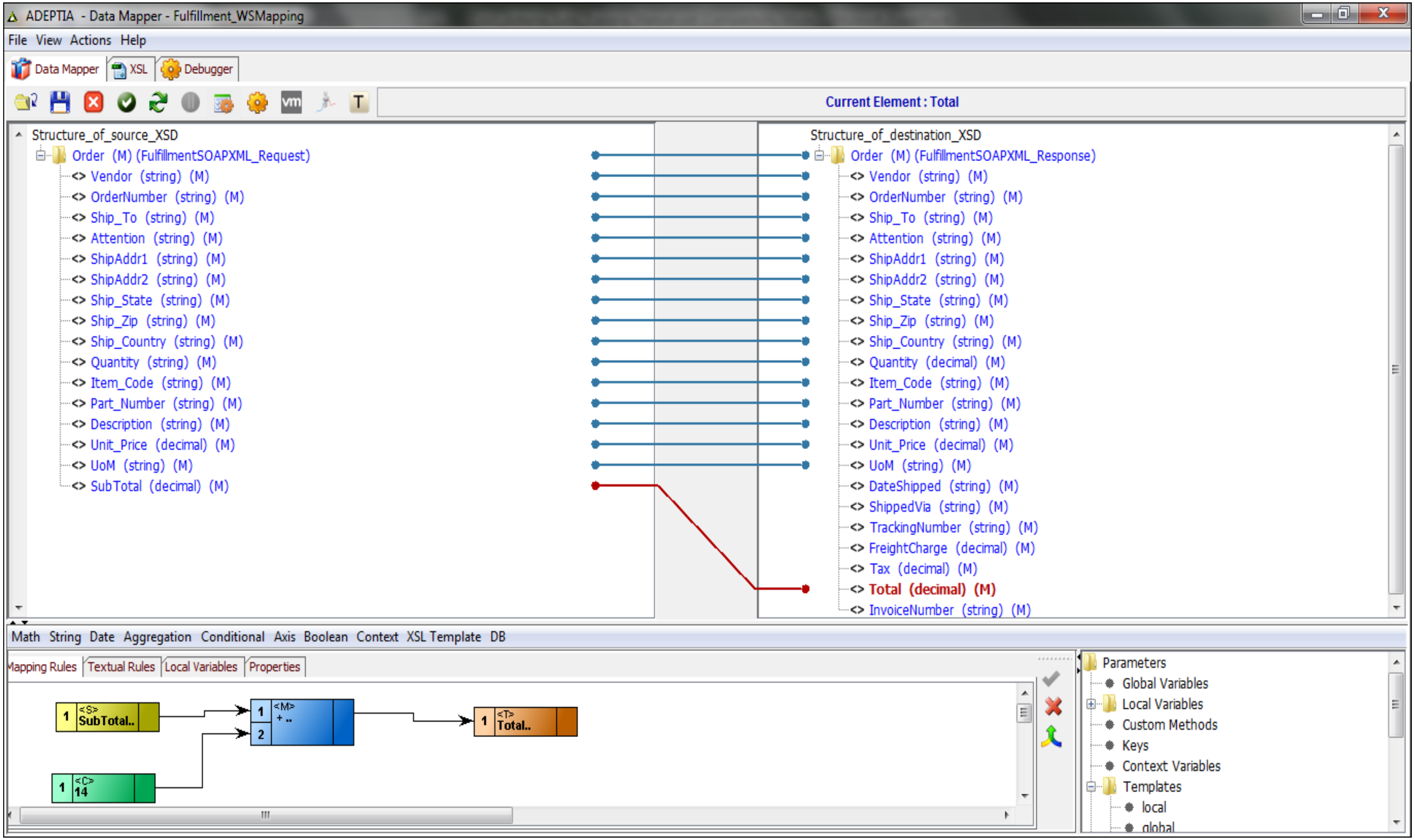
Exercise: Order Fulfillment Process (Create Excel to Request XML map)


ADEPTIA - Data Mapper - Fulfillment_ExcelToRequestXML
File View Actions Help
Data Mapper
XSL
Debugger
No Element Selected
Structure_of_source_XSD
Root (OrderFulfillmentExcelSchema)
Record (M)
recordNumber
<> Vendor (string) (M)
<> OrderNumber (string) (M)
<> CostCenter (string)
<> ShipTo (string) (M)
<> Attention (string) (M)
<> ShipAddr1 (string) (M)
<> ShipAddr2 (string) (M)
<> ShipCity (string)
<> ShipState (string) (M)
<> ShipZip (string) (M)
<> ShipCountry (string) (M)
<> Quantity (integer) (M)
<> EPIItemCode (string) (M)
Structure_of_destination_XSD
Order (M) (FulfillmentSOAPXML_Request)
<> Vendor (string) (M)
<> OrderNumber (string) (M)
<> Ship_To (string) (M)
<> Attention (string) (M)
<> ShipAddr1 (string) (M)
<> ShipAddr2 (string) (M)
<> Ship_State (string) (M)
<> Ship_Zip (string) (M)
<> Ship_Country (string) (M)
<> Quantity (string) (M)
<> Item_Code (string) (M)
<> Part_Number (string) (M)
<> Description (string) (M)
<> Unit_Price (decimal) (M)
<> UoM (string) (M)
Math String Date Aggregation Conditional Axis Boolean Context XSL Template DB
Node
Global
Data Viewer
Mapping Rules
Textual Rules
Local Variables
Properties
Parameters
Global Variables
Local Variables
Custom Methods
Keys
Context Variables
Templates
local
global
group
Connection Info
Global Methods
Value Map
global
group

Exercise: Order Fulfillment Process (Create Response XML to Excel map)


ADEPTIA - Data Mapper - Fulfillment_ResponseXMLtoExcel
File View Actions Help
Data Mapper
XSL
Debugger
Current Element : ShipCity
Structure_of_source_XSD
Order (M) (FulfillmentSOAPXML_Response)
<> Vendor (string) (M)
<> OrderNumber (string) (M)
<> Ship_To (string) (M)
<> Attention (string) (M)
<> ShipAddr1 (string) (M)
<> ShipAddr2 (string) (M)
<> Ship_State (string) (M)
<> Ship_Zip (string) (M)
<> Ship_Country (string) (M)
<> Quantity (decimal) (M)
<> Item_Code (string) (M)
<> Part_Number (string) (M)
<> Description (string) (M)
<> Unit_Price (decimal) (M)
<> UoM (string) (M)
<> DateShipped (string) (M)
<> ShippedVia (string) (M)
<> TrackingNumber (string) (M)
<> FreightCharge (decimal) (M)
Structure_of_destination_XSD
Root (OrderFulfillmentExcelSchema)
Record (M)
@ recordNumber
<> Vendor (string) (M)
<> OrderNumber (string) (M)
<> CostCenter (string)
<> ShipTo (string) (M)
<> Attention (string) (M)
<> ShipAddr1 (string) (M)
<> ShipAddr2 (string) (M)
<> ShipCity (string) (M)
<> ShipState (string) (M)
<> ShipZip (string) (M)
<> ShipCountry (string) (M)
<> Quantity (integer) (M)
<> EPIItemCode (string) (M)
<> VendorPartNumber (string) (M)
<> Description (string) (M)
<> UnitPrice (integer) (M)
<> UoM (string) (M)
Math String Date Aggregation Conditional Axis Boolean Context XSL Template DB
Mapping Rules
Textual Rules
Local Variables
Properties
1 <S> Ship_Zip..
1 <T> ShipCity..
Parameters
Global Variables
Local Variables
Custom Methods
Keys
Context Variables
Templates
local
global
group

Exercise: Order Fulfillment Process (Create Request XML to Response XML Web Service Mapping)



Structure_of_source_XSD

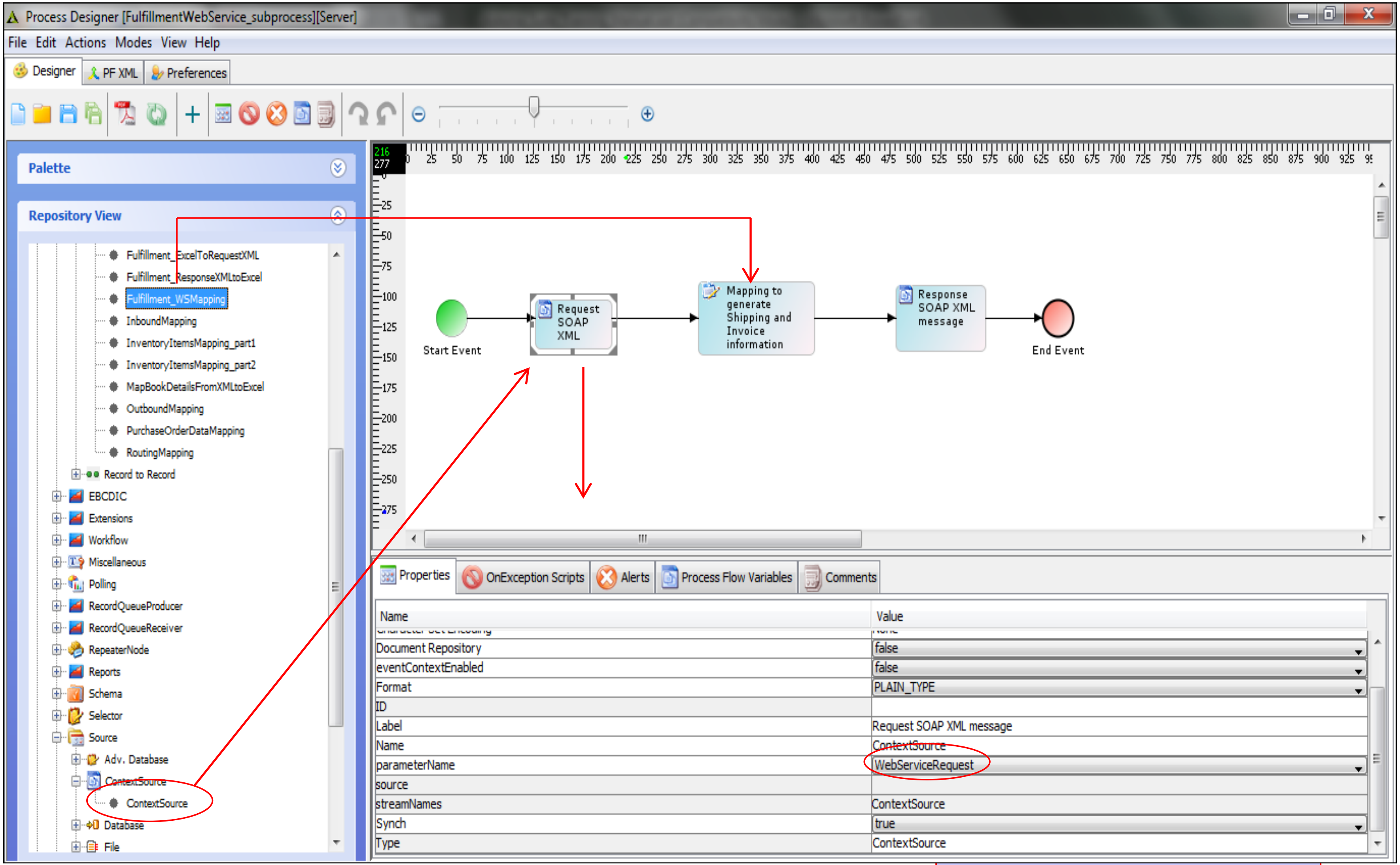
- Order (M) (FulfillmentSOAPXML_Request)
  - <> Vendor (string) (M)
  - <> OrderNumber (string) (M)
  - <> Ship_To (string) (M)
  - <> Attention (string) (M)
  - <> ShipAddr1 (string) (M)
  - <> ShipAddr2 (string) (M)
  - <> Ship_State (string) (M)
  - <> Ship_Zip (string) (M)
  - <> Ship_Country (string) (M)
  - <> Quantity (string) (M)
  - <> Item_Code (string) (M)
  - <> Part_Number (string) (M)
  - <> Description (string) (M)
  - <> Unit_Price (decimal) (M)
  - <> UoM (string) (M)
  - <> SubTotal (decimal) (M)

Structure_of_destination_XSD

- Order (M) (FulfillmentSOAPXML_Response)
  - <> Vendor (string) (M)
  - <> OrderNumber (string) (M)
  - <> Ship_To (string) (M)
  - <> Attention (string) (M)
  - <> ShipAddr1 (string) (M)
  - <> ShipAddr2 (string) (M)
  - <> Ship_State (string) (M)
  - <> Ship_Zip (string) (M)
  - <> Ship_Country (string) (M)
  - <> Quantity (decimal) (M)
  - <> Item_Code (string) (M)
  - <> Part_Number (string) (M)
  - <> Description (string) (M)
  - <> Unit_Price (decimal) (M)
  - <> UoM (string) (M)
  - <> DateShipped (string) (M)
  - <> ShippedVia (string) (M)
  - <> TrackingNumber (string) (M)
  - <> FreightCharge (decimal) (M)
  - <> Tax (decimal) (M)
  - <> Total (decimal) (M)
  - <> InvoiceNumber (string) (M)

Refer to Notes below

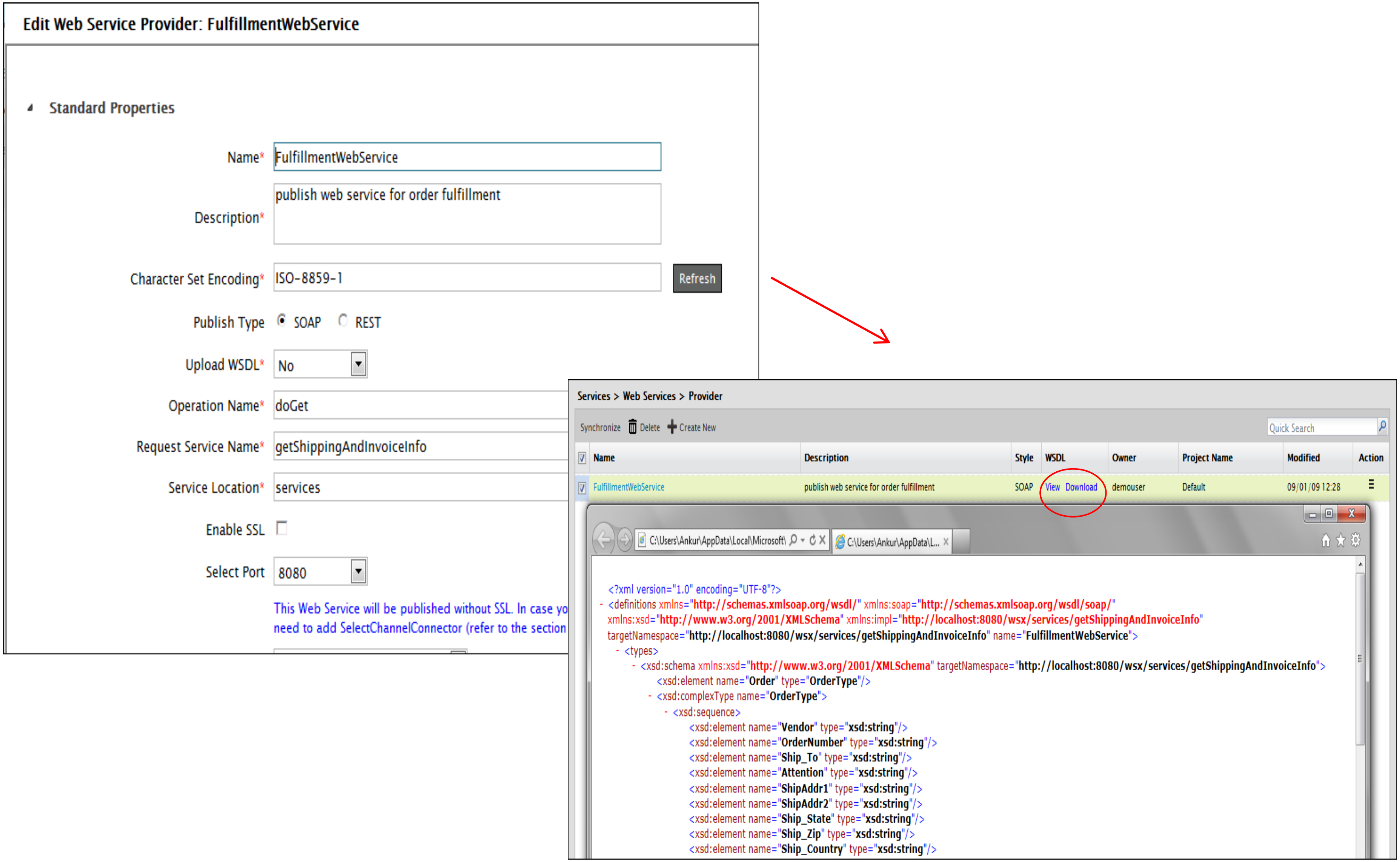
Exercise: Order Fulfillment Process (configure Fulfillment WS sub process)



Name	Value
Document Repository	false
eventContextEnabled	false
Format	PLAIN_TYPE
ID	
Label	Request SOAP XML message
Name	ContextSource
parameterName	WebServiceRequest
source	
streamNames	ContextSource
Synch	true
Type	ContextSource

Refer to Notes below

Exercise: Order Fulfillment Process (Publish Web Service)



Edit Web Service Provider: FulfillmentWebService

Standard Properties

Name*	FulfillmentWebService
Description*	publish web service for order fulfillment
Character Set Encoding*	ISO-8859-1
Publish Type	● SOAP ○ REST
Upload WSDL*	No
Operation Name*	doGet
Request Service Name*	getShippingAndInvoiceInfo
Service Location*	services
Enable SSL	□
Select Port	8080

This Web Service will be published without SSL. In case yo
need to add SelectChannelConnector (refer to the section

Services > Web Services > Provider

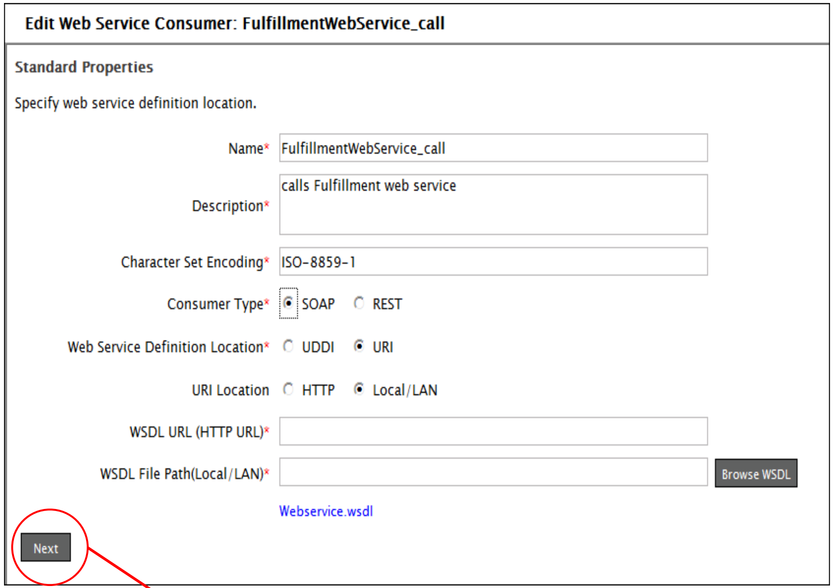
☑	Name	Description	Style	WSDL	Owner	Project Name	Modified	Action
☑	FulfillmentWebService	publish web service for order fulfillment	SOAP	View Download	demouser	Default	09/01/09 12:28	≡

C:\Users\Ankur\AppData\Local\Microsoft\
C:\Users\Ankur\AppData\L...

```
<?xml version="1.0" encoding="UTF-8"?>
- <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:impl="http://localhost:8080/wsx/services/getShippingAndInvoiceInfo"
  targetNamespace="http://localhost:8080/wsx/services/getShippingAndInvoiceInfo" name="FulfillmentWebService">
  - <types>
    - <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://localhost:8080/wsx/services/getShippingAndInvoiceInfo">
        <xsd:element name="Order" type="OrderType"/>
      - <xsd:complexType name="OrderType">
        - <xsd:sequence>
            <xsd:element name="Vendor" type="xsd:string"/>
            <xsd:element name="OrderNumber" type="xsd:string"/>
            <xsd:element name="Ship_To" type="xsd:string"/>
            <xsd:element name="Attention" type="xsd:string"/>
            <xsd:element name="ShipAddr1" type="xsd:string"/>
            <xsd:element name="ShipAddr2" type="xsd:string"/>
            <xsd:element name="Ship_State" type="xsd:string"/>
            <xsd:element name="Ship_Zip" type="xsd:string"/>
            <xsd:element name="Ship_Country" type="xsd:string"/>
```

Refer to Notes below

Exercise: Order Fulfillment Process (create the Web Service Consumer)

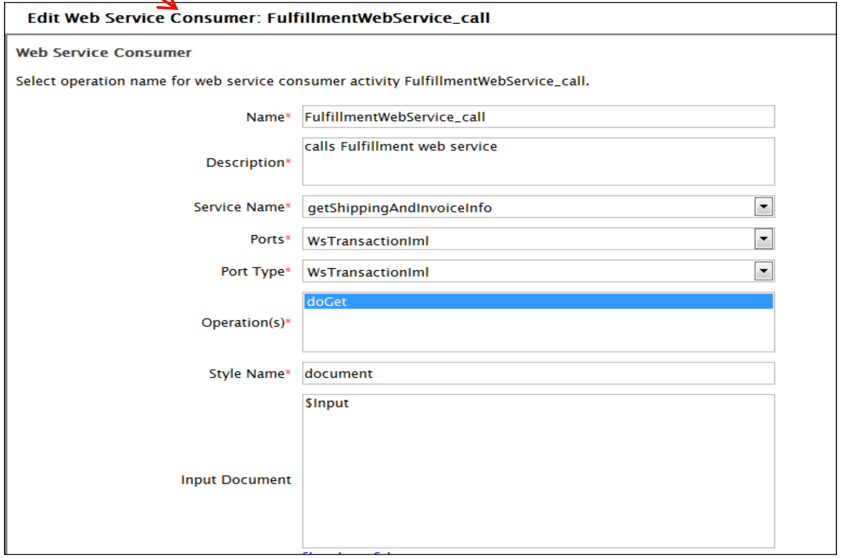


Edit Web Service Consumer: FulfillmentWebService_call

Standard Properties

Specify web service definition location.

Name*	FulfillmentWebService_call
Description*	calls Fulfillment web service
Character Set Encoding*	ISO-8859-1
Consumer Type*	● SOAP ○ REST
Web Service Definition Location*	○ UDDI ● URI
URI Location	○ HTTP ● Local/LAN
WSDL URL (HTTP URL)*	
WSDL File Path(Local/LAN)*	

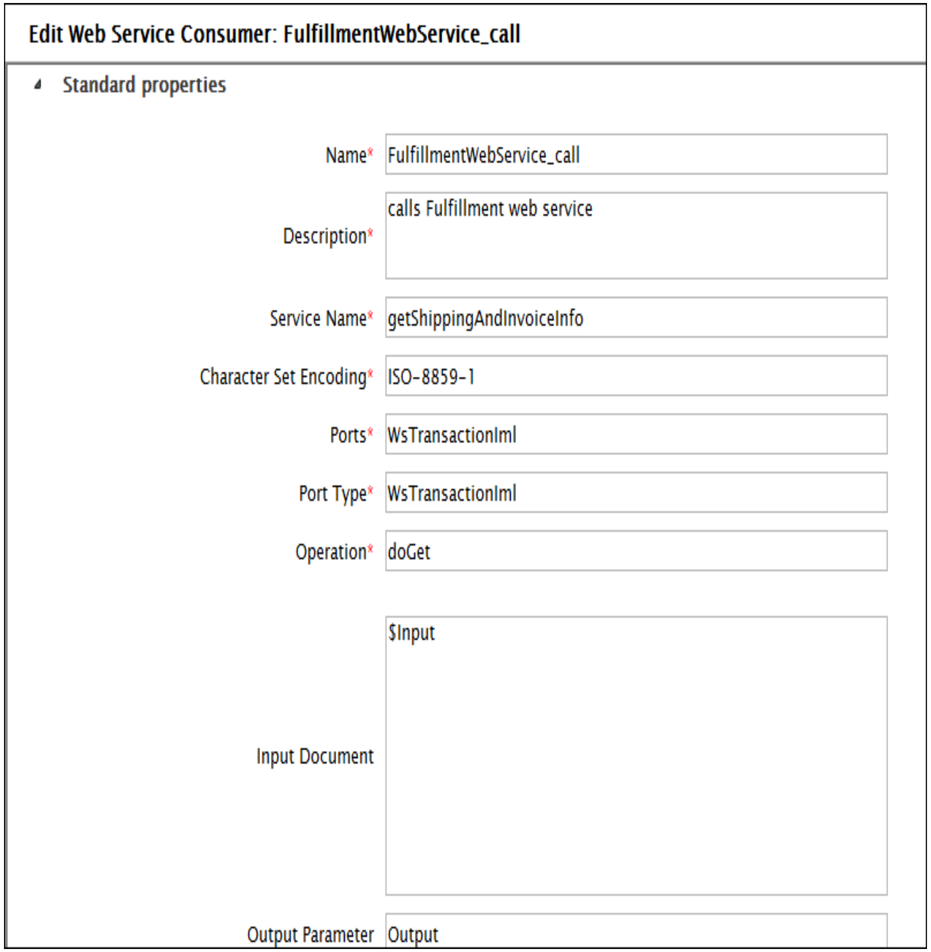


Edit Web Service Consumer: FulfillmentWebService_call

Web Service Consumer

Select operation name for web service consumer activity FulfillmentWebService_call.

Name*	FulfillmentWebService_call
Description*	calls Fulfillment web service
Service Name*	getShippingAndInvoiceInfo
Ports*	WsTransactionImI
Port Type*	WsTransactionImI
Operation(s)*	doGet
Style Name*	document
Input Document	$Input



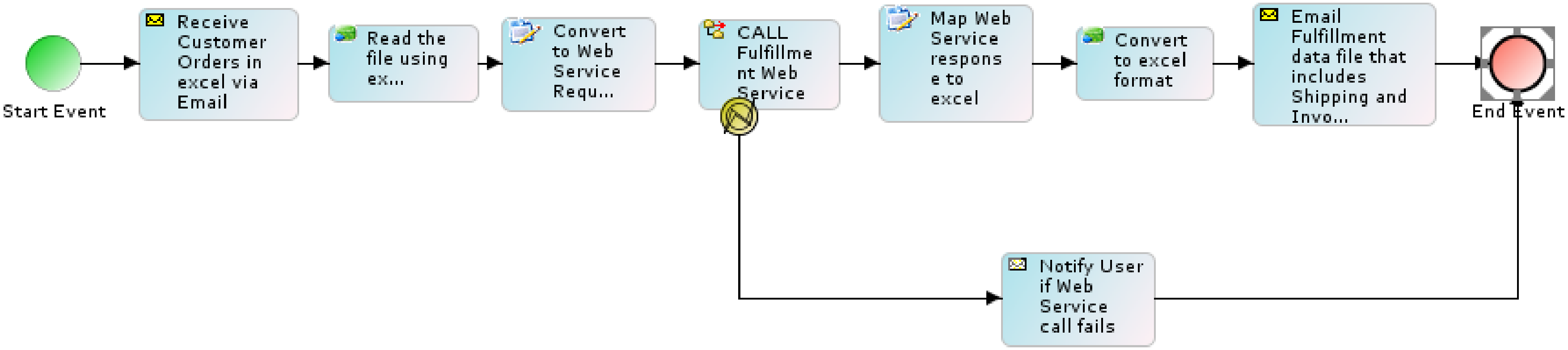
Edit Web Service Consumer: FulfillmentWebService_call

Standard properties

Name*	FulfillmentWebService_call
Description*	calls Fulfillment web service
Service Name*	getShippingAndInvoiceInfo
Character Set Encoding*	ISO-8859-1
Ports*	WsTransactionImI
Port Type*	WsTransactionImI
Operation*	doGet
Input Document	$Input
Output Parameter	Output

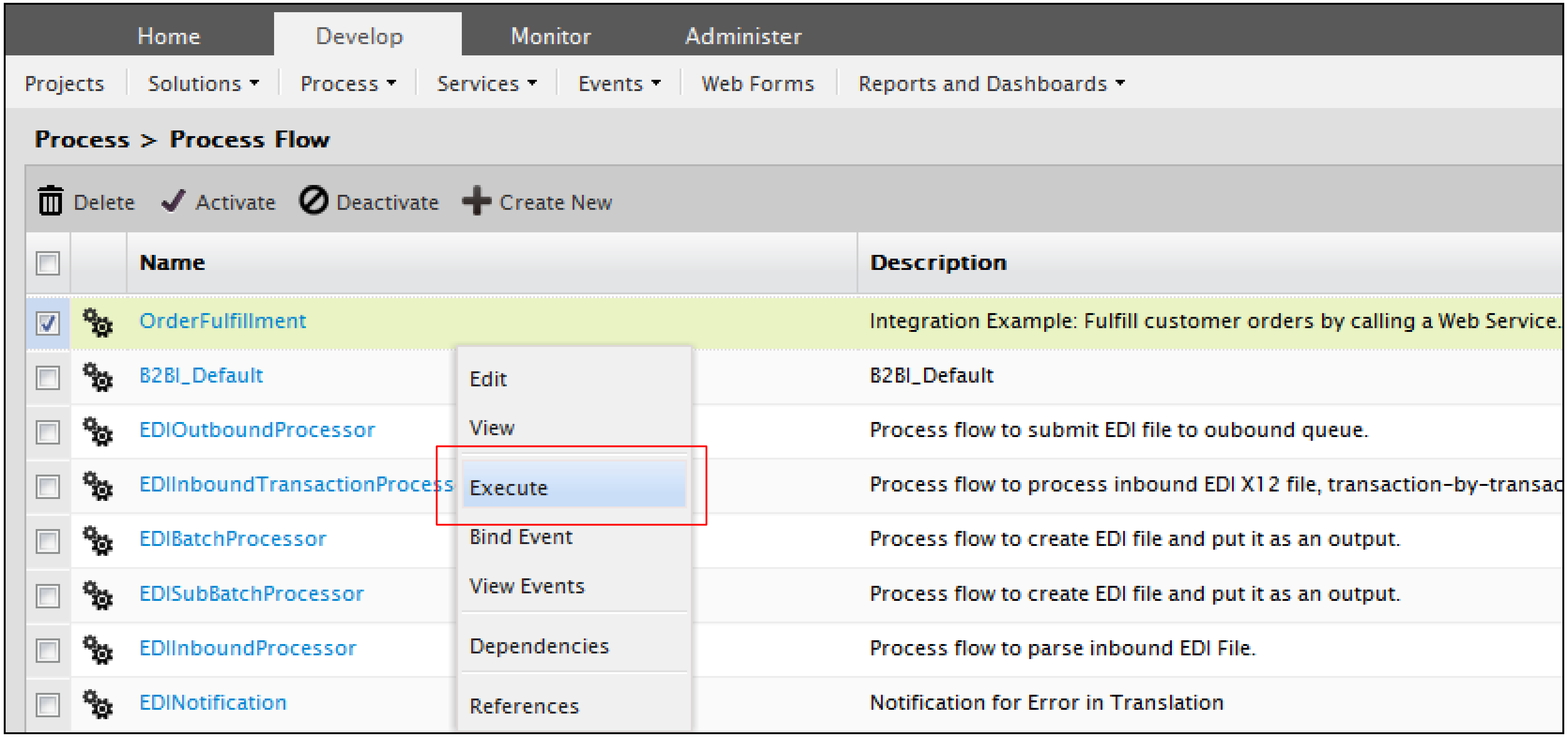
Refer to Notes below

Exercise: Order Fulfillment Process (configure parent process)


Start Event
Receive Customer Orders in excel via Email
Read the file using ex...
Convert to Web Service Requ...
CALL Fulfillment Web Service
Map Web Service response to excel
Convert to excel format
Email Fulfillment data file that includes Shipping and Invo...
End Event
Notify User if Web Service call fails

Refer to Notes below

Exercise: Order Fulfillment (deploy process flow)



□		Name	Description
☑		OrderFulfillment	Integration Example: Fulfill customer orders by calling a Web Service.
□		B2BI_Default	B2BI_Default
□		EDIOutboundProcessor	Process flow to submit EDI file to oubound queue.
□		EDIInboundTransactionProcess	Process flow to process inbound EDI X12 file, transaction-by-transac
□		EDIBatchProcessor	Process flow to create EDI file and put it as an output.
□		EDISubBatchProcessor	Process flow to create EDI file and put it as an output.
□		EDIInboundProcessor	Process flow to parse inbound EDI File.
□		EDINotification	Notification for Error in Translation

Exercise: Inventory Items Processing
(With correction and re-submission through Workflow task)

Exercise: Inventory Items Process

Inventory Items Process requires conversion of CSV source data file into XML. Source data file is fetched from a FTP location. If some of the critical fields are missing, the data record will be routed to a Workflow task for correction and re-submission. Correct source data is converted into an XML output file.

Design:

- Get Inventory Items CSV data file from source FTP
- Apply rules to check for critical fields (to be done in Mapping)
- Route incorrect data to Workflow task for correction and re-submission
- Convert valid data to XML

Configure:

- Configure the process by creating below activities
  - Create FTP Source
  - Create Text Schema for the CSV file
  - Create XML Schema for the output file
  - Create Workflow Task form
  - Create Mapping
  - Create File Target

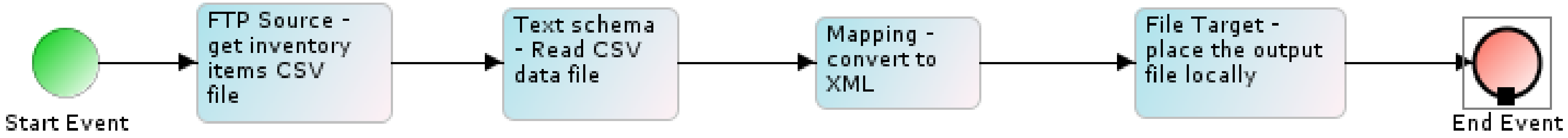
Deploy:

- Create FTP Event
- Register the FTP Event with Inventory Item process flow
- Activate FTP Event

Refer to Notes below

Exercise: Inventory Items Process – Part 1 (design)

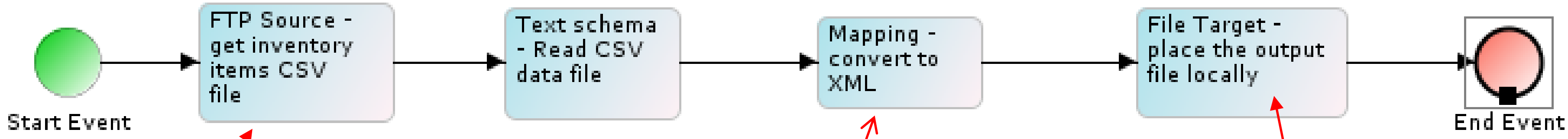
Part 1: First create a simple CSV to XML process. This will take the file and convert it into XML.

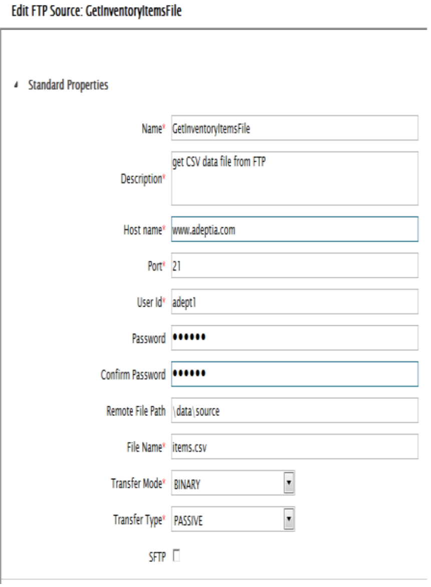

Start Event
FTP Source - get inventory items CSV file
Text schema - Read CSV data file
Mapping - convert to XML
File Target - place the output file locally
End Event

Refer to Notes below

Exercise: Inventory Items Process – Part 1 (configure)

Part 1: First create a simple CSV to XML process. This will take the file and convert it into XML.

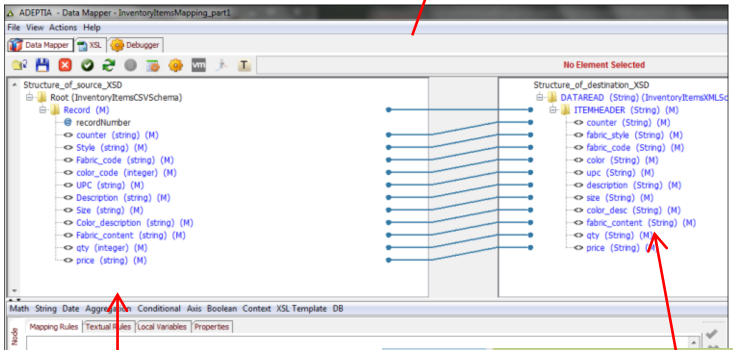

Start Event
FTP Source - get inventory items CSV file
Text schema - Read CSV data file
Mapping - convert to XML
File Target - place the output file locally
End Event



Edit FTP Source: GetInventoryItemsFile

Standard Properties

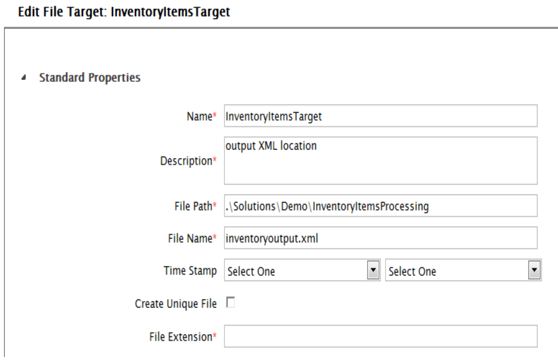
Name*	GetInventoryItemsFile
Description*	get CSV data file from FTP
Host name*	www.adeptia.com
Port*	21
User Id*	adept1
Password*	••••••
Confirm Password*	••••••
Remote File Path	\data\source
File Name*	items.csv
Transfer Mode*	BINARY
Transfer Type*	PASSIVE
SFTP	□



ADEPTIA - Data Mapper - InventoryItemsMapping_part1

Structure_of_source_XSD
- Root (InventoryItemsCSVSchema)
  - Record (M)
    - @ recordNumber
    - <> counter (string) (M)
    - <> Style (string) (M)
    - <> Fabric_code (string) (M)
    - <> color_code (integer) (M)
    - <> UPC (string) (M)
    - <> Description (string) (M)
    - <> Size (string) (M)
    - <> Color_description (string) (M)
    - <> Fabric_content (string) (M)
    - <> qty (integer) (M)
    - <> price (string) (M)

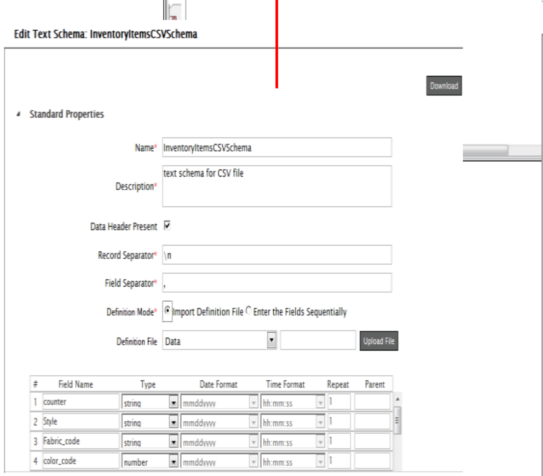
Structure_of_destination_XSD
- DATAREAD (String) (InventoryItemsXMLSc
  - ITEMHEADER (String) (M)
    - <> counter (String) (M)
    - <> fabric_style (String) (M)
    - <> fabric_code (String) (M)
    - <> color (String) (M)
    - <> upc (String) (M)
    - <> description (String) (M)
    - <> size (String) (M)
    - <> color_desc (String) (M)
    - <> fabric_content (String) (M)
    - <> qty (String) (M)
    - <> price (String) (M)



Edit File Target: InventoryItemsTarget

Standard Properties

Name*	InventoryItemsTarget
Description*	output XML location
File Path*	.\Solutions\Demo\InventoryItemsProcessing
File Name*	inventoryoutput.xml
Time Stamp	Select One / Select One
Create Unique File	□
File Extension*	

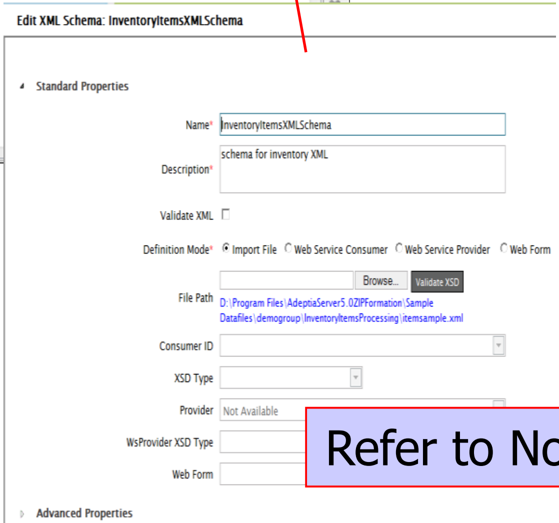


Edit Text Schema: InventoryItemsCSVSchema

Standard Properties

Name*	InventoryItemsCSVSchema
Description*	text schema for CSV file
Data Header Present	☑
Record Separator*	\n
Field Separator*	,
Definition Mode*	● Import Definition File ○ Enter the Fields Sequentially
Definition File	Data

#	Field Name	Type	Date Format	Time Format	Repeat	Parent
1	counter	string	mmddyyyy	hh:mm:ss	1	
2	Style	string	mmddyyyy	hh:mm:ss	1	
3	Fabric_code	string	mmddyyyy	hh:mm:ss	1	
4	color_code	number	mmddyyyy	hh:mm:ss	1	



Edit XML Schema: InventoryItemsXMLSchema

Standard Properties

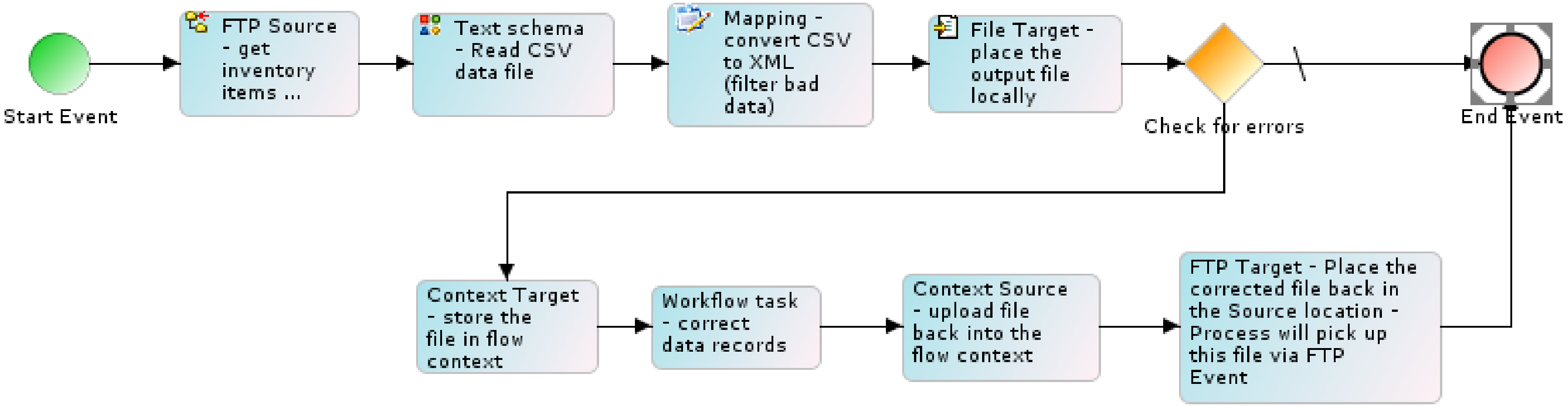
Name*	InventoryItemsXMLSchema
Description*	schema for inventory XML
Validate XML	□
Definition Mode*	● Import File ○ Web Service Consumer ○ Web Service Provider ○ Web Form
File Path	D:\Program Files\AdeptiaServer5.0ZIPFormation\Sample Datafiles\demogroup\InventoryItemsProcessing\itemsample.xml
Consumer ID	
XSD Type	
Provider	Not Available
WsProvider XSD Type	
Web Form	

Advanced Properties

Refer to Notes below

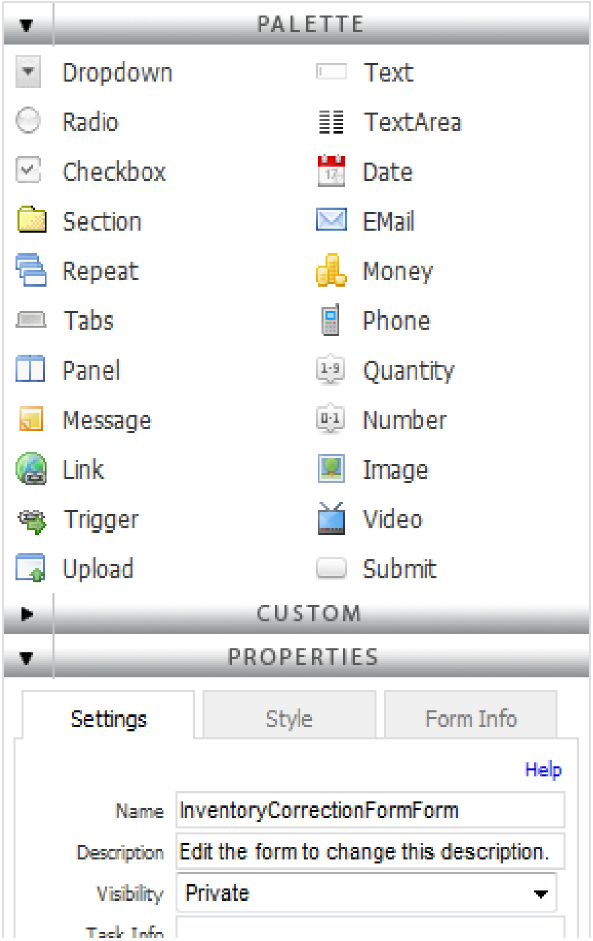
Exercise: Inventory Items Process – Part 2 (design change)

Part 2: Edit Mapping to apply validation rules. Add workflow task for correction and re-submission.


Start Event
FTP Source - get inventory items ...
Text schema - Read CSV data file
Mapping - convert CSV to XML (filter bad data)
File Target - place the output file locally
Check for errors
End Event
Context Target - store the file in flow context
Workflow task - correct data records
Context Source - upload file back into the flow context
FTP Target - Place the corrected file back in the Source location - Process will pick up this file via FTP Event

Refer to Notes below

Exercise: Inventory Items Process – Part 2 (create Web Form)

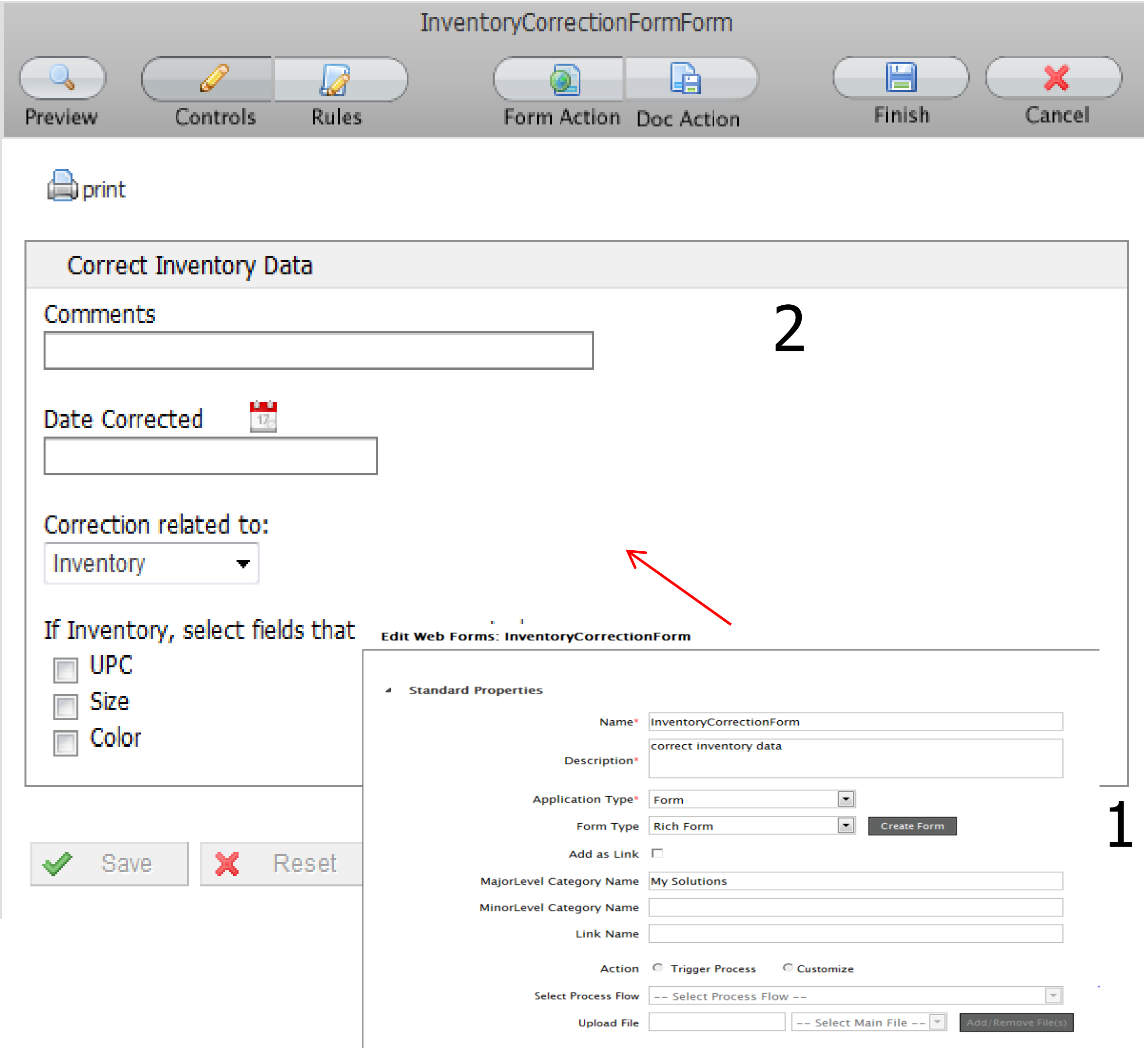


Name InventoryCorrectionFormForm

Description Edit the form to change this description.

Visibility Private

Task Info



Correct Inventory Data

Comments

Date Corrected

Correction related to:

Inventory

If Inventory, select fields that

□ UPC

□ Size

□ Color

Edit Web Forms: InventoryCorrectionForm

Standard Properties

Name* InventoryCorrectionForm

Description* correct inventory data

Application Type* Form

Form Type Rich Form

Add as Link □

MajorLevel Category Name My Solutions

MinorLevel Category Name

Link Name

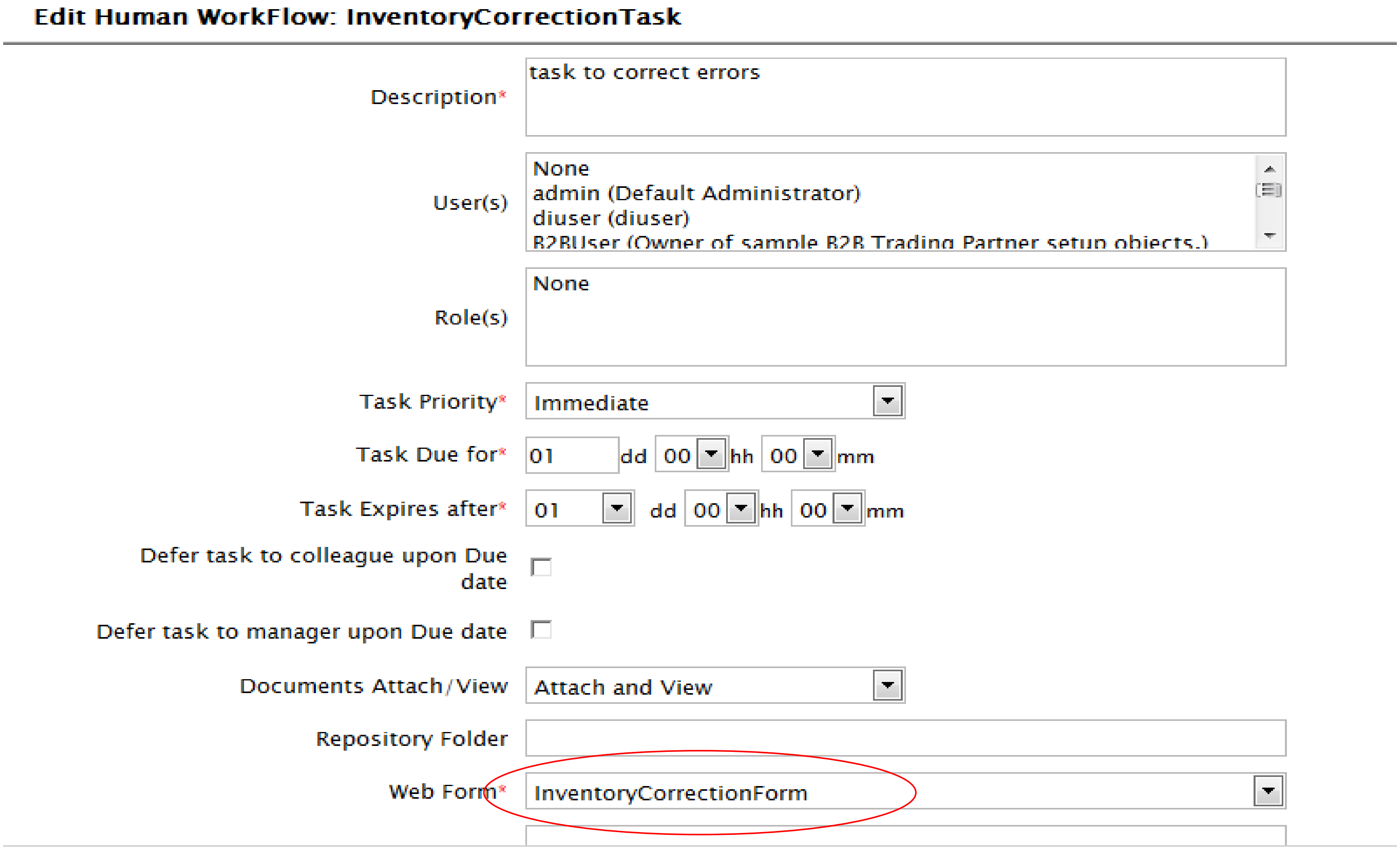
Action ○ Trigger Process ○ Customize

Select Process Flow -- Select Process Flow --

Upload File -- Select Main File --

Refer to Notes below

Exercise: Inventory Items Process – Part 2 (create Workflow task)

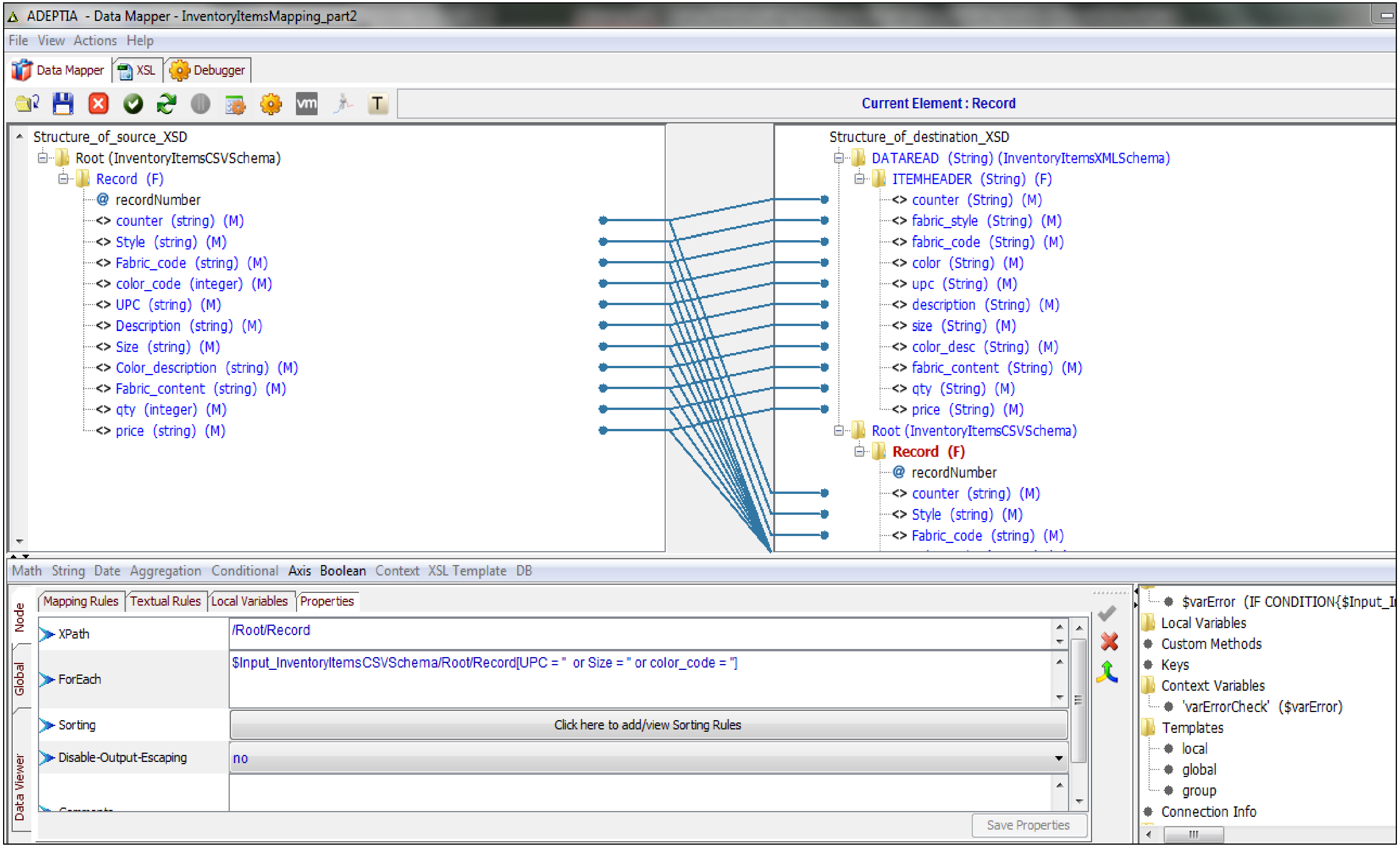


Edit Human WorkFlow: InventoryCorrectionTask

Description*	task to correct errors
User(s)	None admin (Default Administrator) diuser (diuser) B2BUser (Owner of sample B2B Trading Partner setup objects.)
Role(s)	None
Task Priority*	Immediate ▾
Task Due for*	01 dd 00 ▾ hh 00 ▾ mm
Task Expires after*	01 ▾ dd 00 ▾ hh 00 ▾ mm
Defer task to colleague upon Due date	□
Defer task to manager upon Due date	□
Documents Attach/View	Attach and View ▾
Repository Folder	
Web Form*	InventoryCorrectionForm ▾

Refer to Notes below

Exercise: Inventory Items Process – Part 2 (edit Mapping)


ADEPTIA - Data Mapper - InventoryItemsMapping_part2
File View Actions Help
Data Mapper
XSL
Debugger
Current Element : Record
Structure_of_source_XSD
Root (InventoryItemsCSVSchema)
Record (F)
@ recordNumber
<> counter (string) (M)
<> Style (string) (M)
<> Fabric_code (string) (M)
<> color_code (integer) (M)
<> UPC (string) (M)
<> Description (string) (M)
<> Size (string) (M)
<> Color_description (string) (M)
<> Fabric_content (string) (M)
<> qty (integer) (M)
<> price (string) (M)
Structure_of_destination_XSD
DATAREAD (String) (InventoryItemsXMLSchema)
ITEMHEADER (String) (F)
<> counter (String) (M)
<> fabric_style (String) (M)
<> fabric_code (String) (M)
<> color (String) (M)
<> upc (String) (M)
<> description (String) (M)
<> size (String) (M)
<> color_desc (String) (M)
<> fabric_content (String) (M)
<> qty (String) (M)
<> price (String) (M)
Root (InventoryItemsCSVSchema)
Record (F)
@ recordNumber
<> counter (string) (M)
<> Style (string) (M)
<> Fabric_code (string) (M)
Math String Date Aggregation Conditional Axis Boolean Context XSL Template DB
Node
Global
Data Viewer
Mapping Rules
Textual Rules
Local Variables
Properties
Click here to add/view Sorting Rules
Save Properties
$varError (IF CONDITION{$Input_I
Local Variables
Custom Methods
Keys
Context Variables
'varErrorCheck' ($varError)
Templates
local
global
group
Connection Info

XPath: /Root/Record

ForEach: $Input_InventoryItemsCSVSchema/Root/Record[UPC = '' or Size = '' or color_code = '']

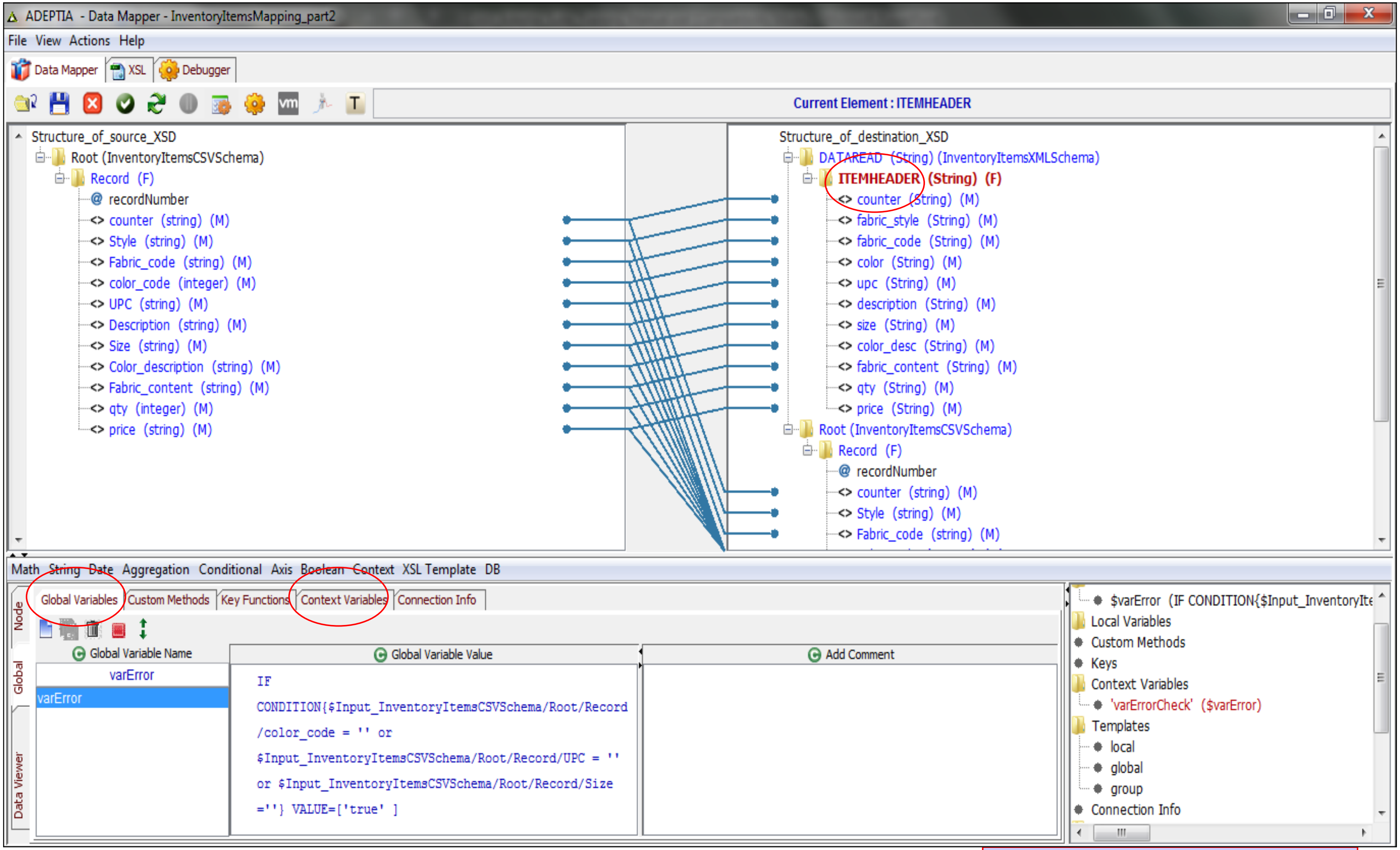
Sorting

Disable-Output-Escaping: no

Comments

Refer to Notes below

Exercise: Inventory Items Process – Part 2 (edit Mapping)



ADEPTIA - Data Mapper - InventoryItemsMapping_part2

Current Element : ITEMHEADER

Structure_of_source_XSD
- Root (InventoryItemsCSVSchema)
  - Record (F)
    - @ recordNumber
    - <> counter (string) (M)
    - <> Style (string) (M)
    - <> Fabric_code (string) (M)
    - <> color_code (integer) (M)
    - <> UPC (string) (M)
    - <> Description (string) (M)
    - <> Size (string) (M)
    - <> Color_description (string) (M)
    - <> Fabric_content (string) (M)
    - <> qty (integer) (M)
    - <> price (string) (M)

Structure_of_destination_XSD
- DATAREAD (String) (InventoryItemsXMLSchema)
  - ITEMHEADER (String) (F)
    - <> counter (String) (M)
    - <> fabric_style (String) (M)
    - <> fabric_code (String) (M)
    - <> color (String) (M)
    - <> upc (String) (M)
    - <> description (String) (M)
    - <> size (String) (M)
    - <> color_desc (String) (M)
    - <> fabric_content (String) (M)
    - <> qty (String) (M)
    - <> price (String) (M)
- Root (InventoryItemsCSVSchema)
  - Record (F)
    - @ recordNumber
    - <> counter (string) (M)
    - <> Style (string) (M)
    - <> Fabric_code (string) (M)

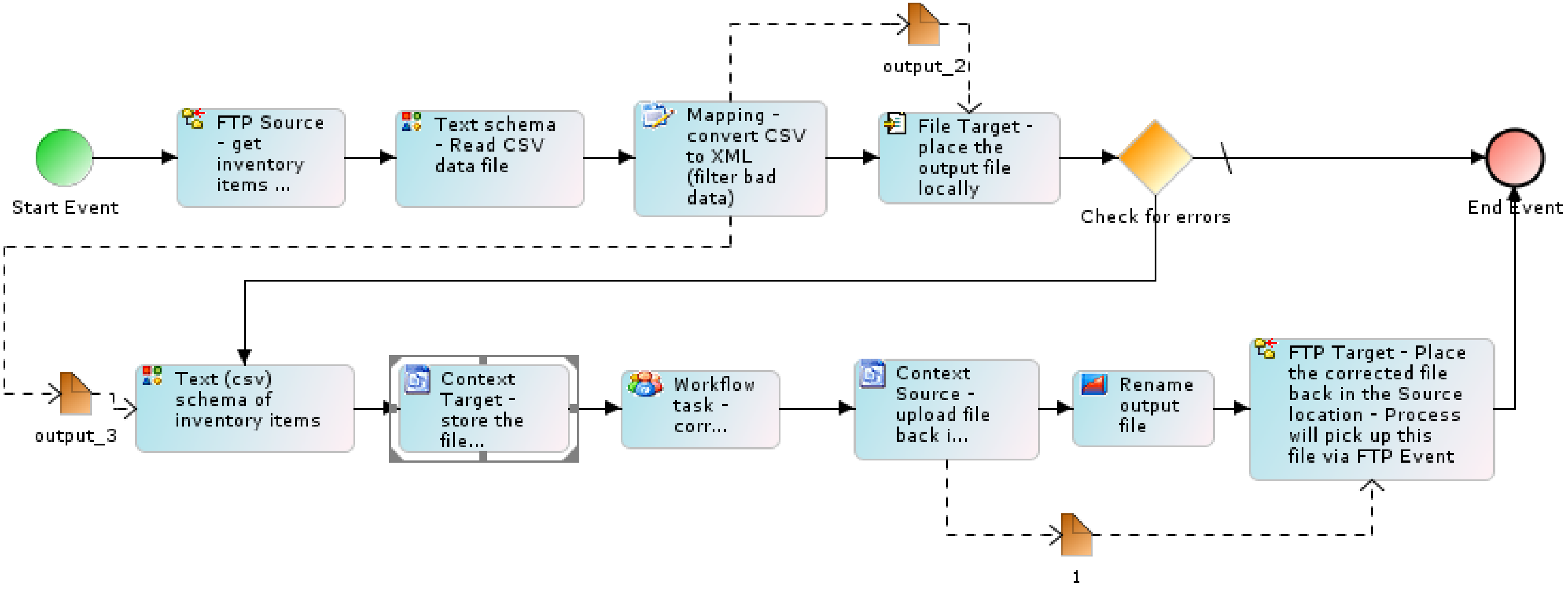
Global Variable Name	Global Variable Value	Add Comment
varError	IF CONDITION{$Input_InventoryItemsCSVSchema/Root/Record/color_code = '' or $Input_InventoryItemsCSVSchema/Root/Record/UPC = '' or $Input_InventoryItemsCSVSchema/Root/Record/Size =''} VALUE=['true']	

```
IF
CONDITION{$Input_InventoryItemsCSVSchema/Root/Record
/color_code = '' or
$Input_InventoryItemsCSVSchema/Root/Record/UPC = ''
or $Input_InventoryItemsCSVSchema/Root/Record/Size
=''} VALUE=['true' ]
```

- $varError (IF CONDITION{$Input_InventoryIte
- Local Variables
- Custom Methods
- Keys
- Context Variables
  - 'varErrorCheck' ($varError)
- Templates
  - local
  - global
  - group
- Connection Info

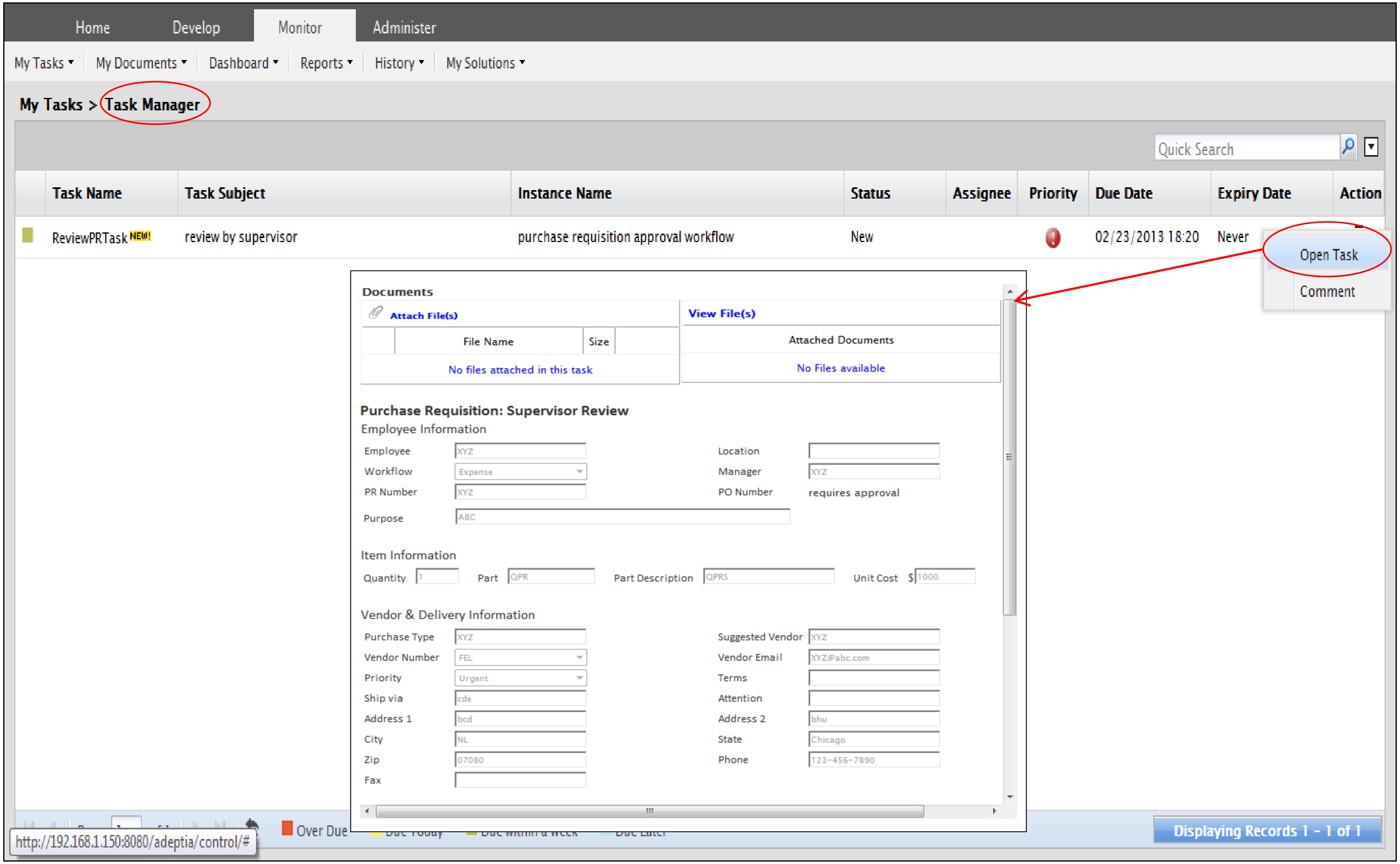
Refer to Notes below

Exercise: Inventory Items Process – Part 2 (configure)


output_2
Start Event
FTP Source - get inventory items ...
Text schema - Read CSV data file
Mapping - convert CSV to XML (filter bad data)
File Target - place the output file locally
Check for errors
End Event
output_3
1
Text (csv) schema of inventory items
Context Target - store the file...
Workflow task - corr...
Context Source - upload file back i...
Rename output file
FTP Target - Place the corrected file back in the Source location - Process will pick up this file via FTP Event
1

Refer to Notes below

Exercise: Inventory Items Process – Part 2 (Workflow Task at run-time)



My Tasks > Task Manager

	Task Name	Task Subject	Instance Name	Status	Assignee	Priority	Due Date	Expiry Date	Action
🟩	ReviewPRTask NEW!	review by supervisor	purchase requisition approval workflow	New		●	02/23/2013 18:20	Never	

Documents

	File Name	Size	
No files attached in this task			

Attached Documents
No Files available

Purchase Requisition: Supervisor Review

Employee Information

Employee	XYZ	Location	
Workflow	Expense	Manager	XYZ
PR Number	XYZ	PO Number	requires approval
Purpose	ABC		

Item Information

Quantity	1	Part	QPR	Part Description	QPRS	Unit Cost	$ 1000

Vendor & Delivery Information

Purchase Type	XYZ	Suggested Vendor	XYZ
Vendor Number	FEL	Vendor Email	XYZ@abc.com
Priority	Urgent	Terms	
Ship via	cde	Attention	
Address 1	bcd	Address 2	bhu
City	NL	State	Chicago
Zip	07080	Phone	123-456-7890
Fax			

Refer to Notes below

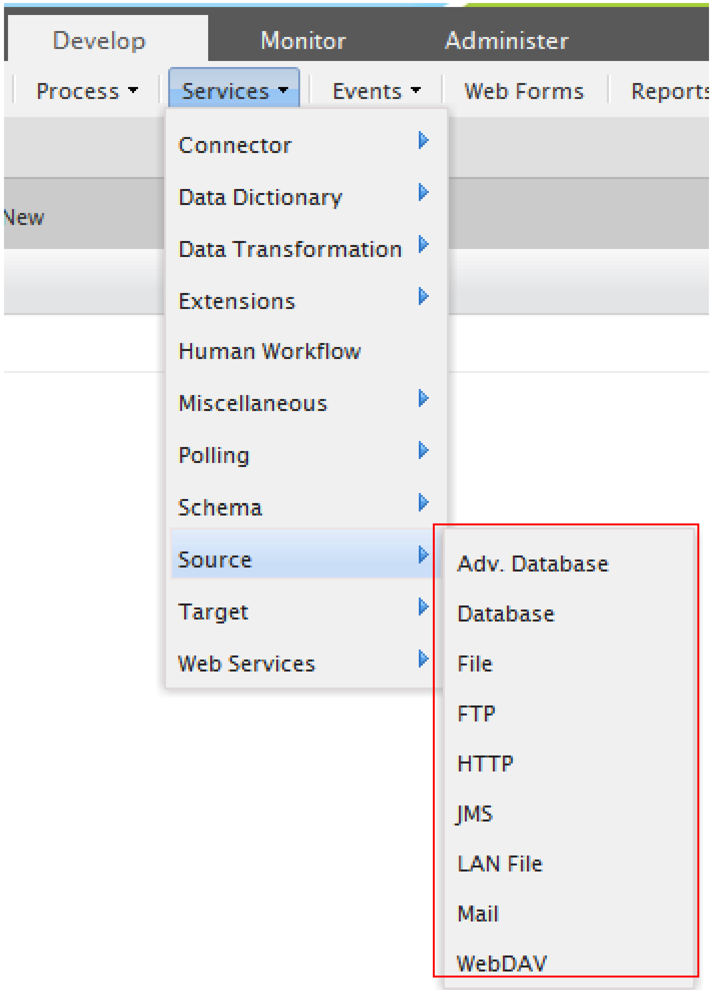
Services

Adeptia Services

Service Categories	Features and Functionality
Transport: Source and Target	FTP, HTTP/S, LAN, Email, JMS, JCA, Databases
Meta-Data: Schema	XML, Text, Fixed-length/Positional, Excel
Data Transformation: Mapping	Math / String / Conditional functions, Database and XML file lookup
Triggers and Events	Polling: Scheduler, Database tables/rows/fields, Files, User response, Email On Demand: Servlet, HTTP, JMS
Notifications	Email
Web Services	Consumer and Publisher
Workflow	Interact with process flows, upload-download documents, take decisions and direct flow of transactions.

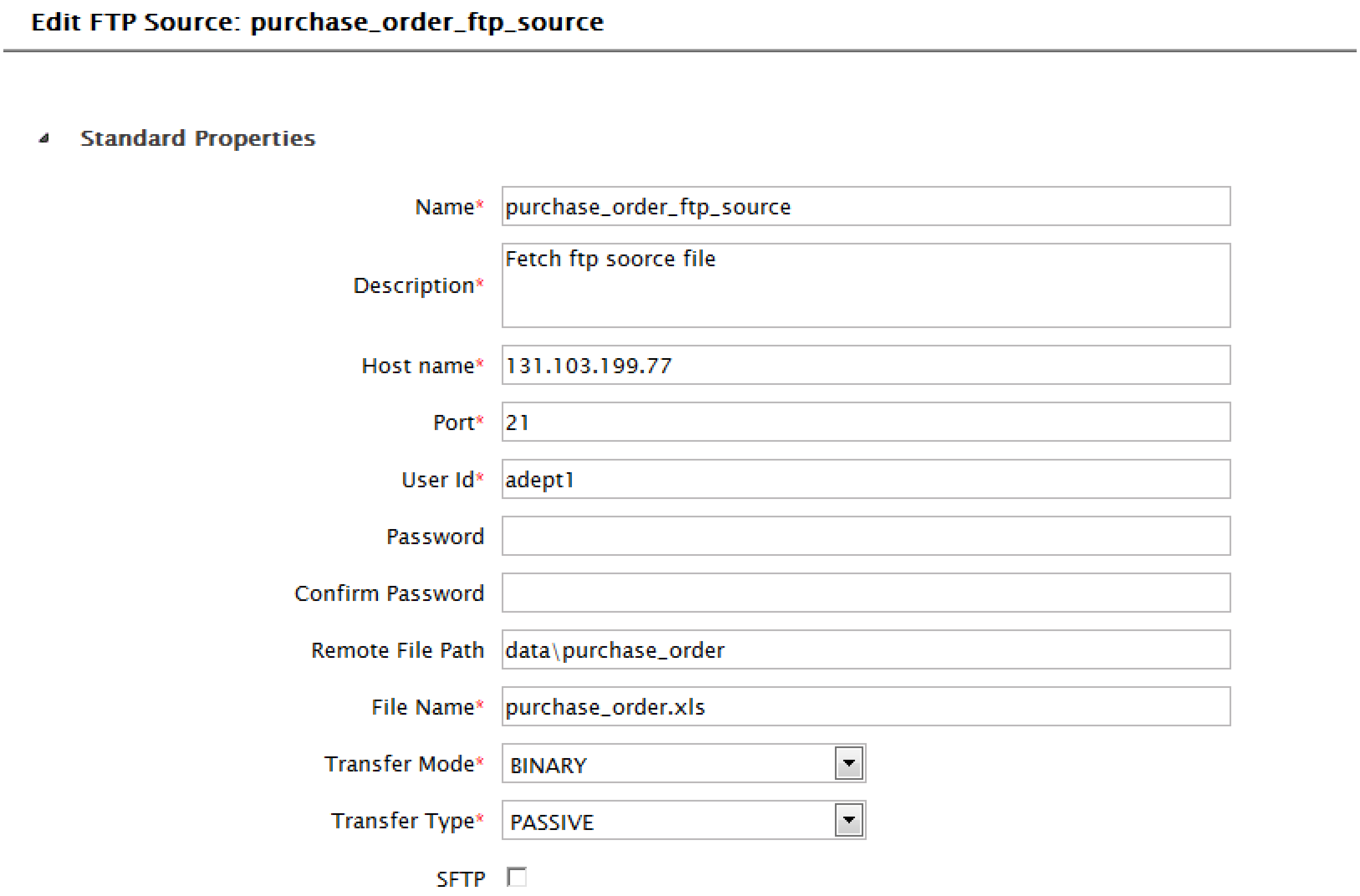
Adeptia Services: What do they do?

Service Categories	What does it do?
Transport: Source and Target	Source: Receives data using the specified protocol and outputs it as a data stream. Target: Takes incoming data stream and delivers it to the target destination using specified protocol.
Meta-Data: Schema	Parses incoming data stream and converts it to intermediate XML (iXML) which is output as a data stream. Vice versa if the incoming stream is iXML.
Data Transformation: Mapping	Takes iXML incoming data stream and applies the mapping rules and outputs iXML data stream.
Triggers and Events	Real-time or scheduler based. Triggers are used to start a process flow. Event is used in a process flow to wait asynchronously.
Notifications	Send Email alert.
Web Services	Consumer: Use an external web service Publisher: Expose a process flow as a web service
Workflow	Stops the execution of a process flow, sends an email to a business user and waits for that user to complete a task related to the flow. Waiting task is shown in the Task Manager.



Source Services

FTP Source

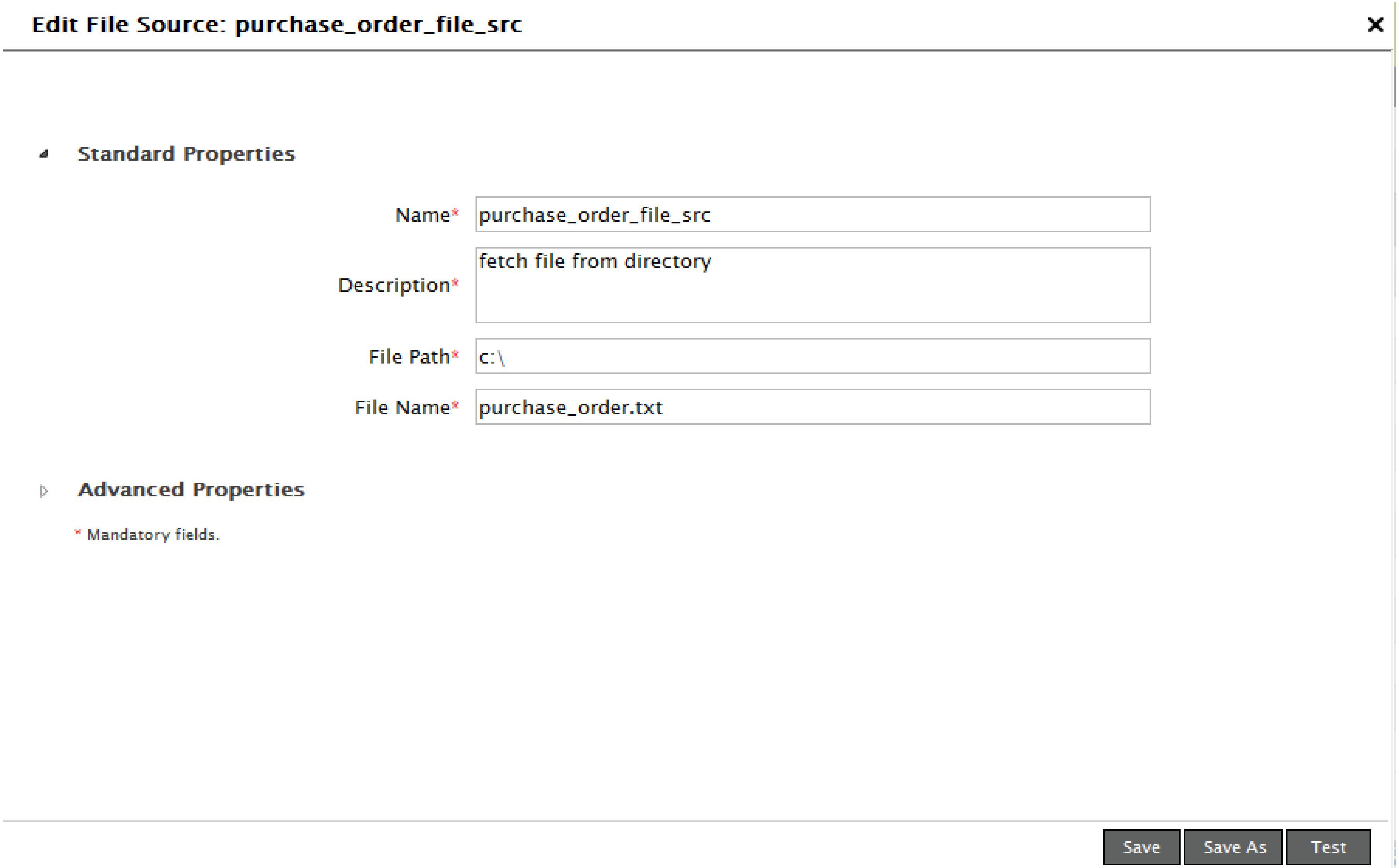


Edit FTP Source: purchase_order_ftp_source

Standard Properties

Name*	purchase_order_ftp_source
Description*	Fetch ftp soorce file
Host name*	131.103.199.77
Port*	21
User Id*	adept1
Password	
Confirm Password	
Remote File Path	data\purchase_order
File Name*	purchase_order.xls
Transfer Mode*	BINARY
Transfer Type*	PASSIVE
SFTP	☐

File Source



Edit File Source: purchase_order_file_src

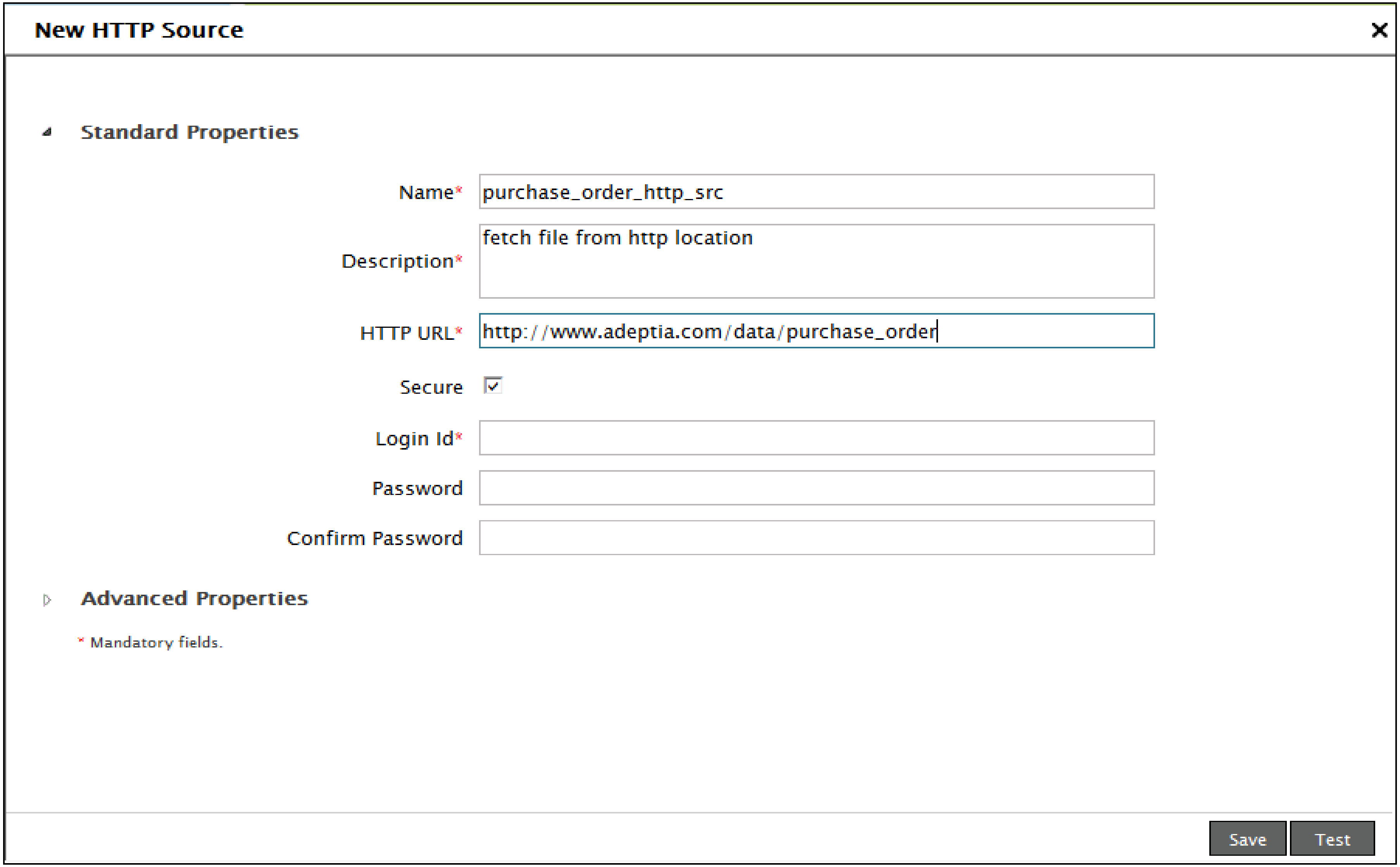
Standard Properties

Name*	purchase_order_file_src
Description*	fetch file from directory
File Path*	c:\
File Name*	purchase_order.txt

Advanced Properties

* Mandatory fields.

HTTP Source



New HTTP Source

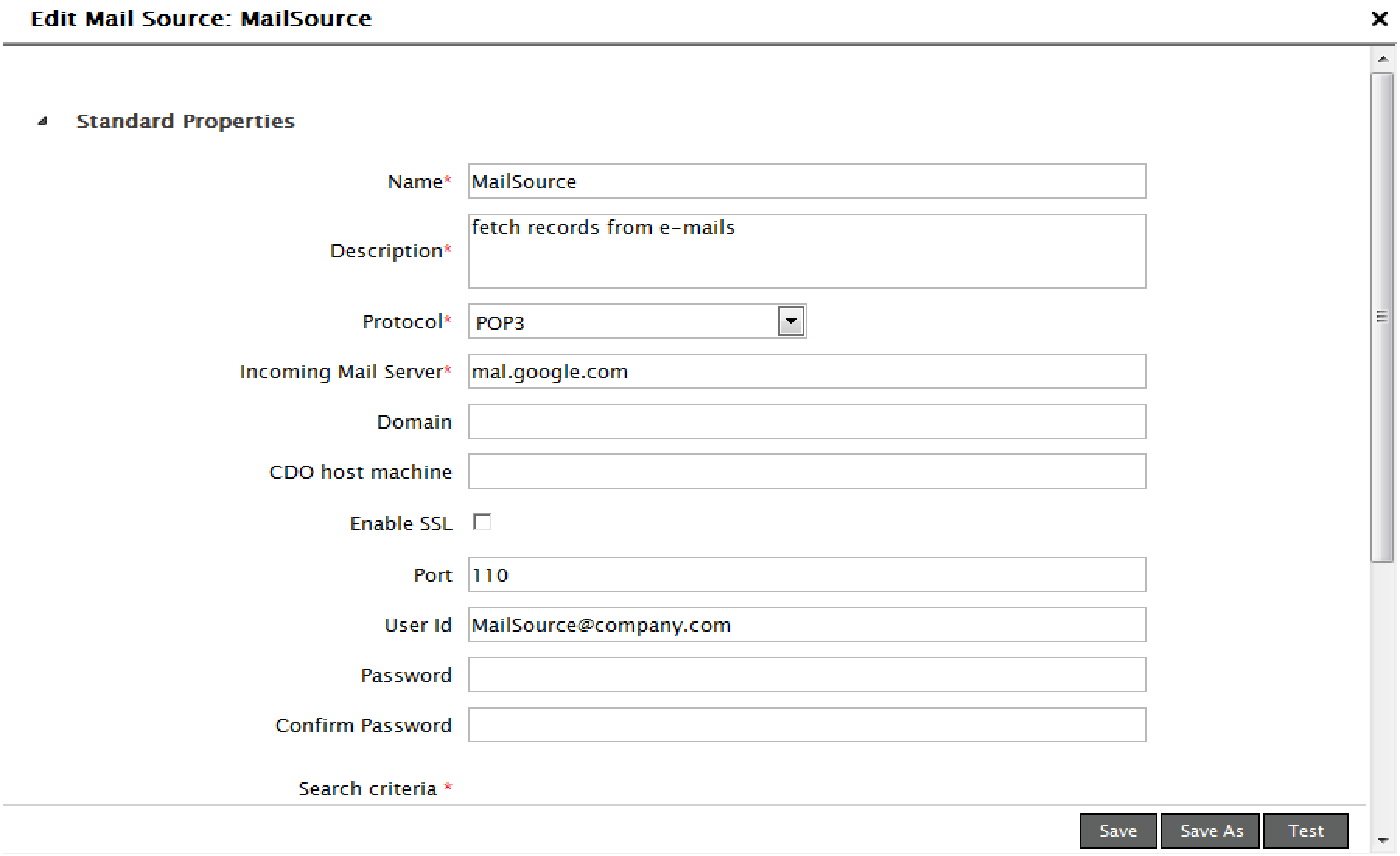
Standard Properties

Name*	purchase_order_http_src
Description*	fetch file from http location
HTTP URL*	http://www.adeptia.com/data/purchase_order
Secure	☑
Login Id*	
Password	
Confirm Password	

Advanced Properties

* Mandatory fields.

Email Source

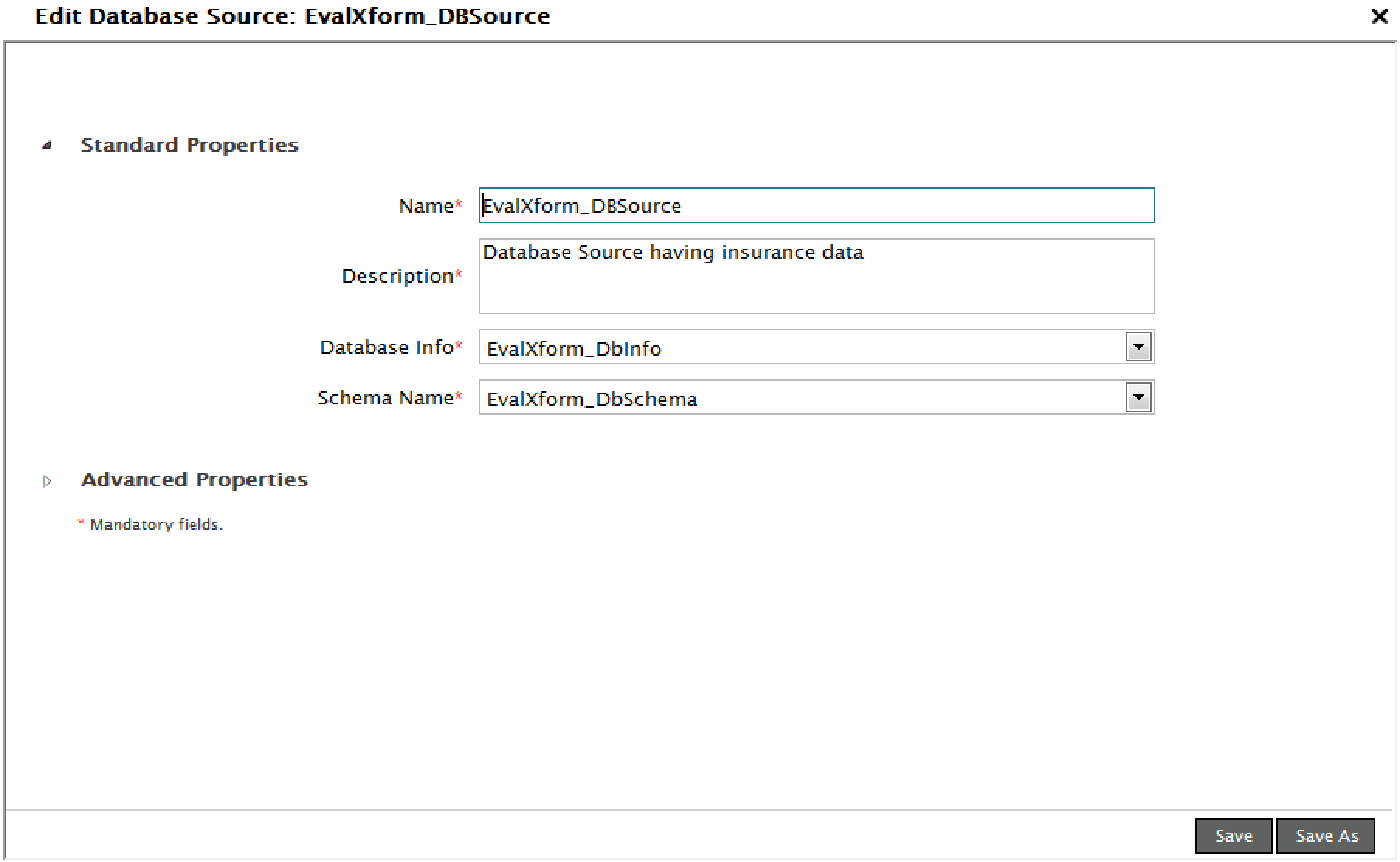


Edit Mail Source: MailSource

Standard Properties

Name*	MailSource
Description*	fetch records from e-mails
Protocol*	POP3
Incoming Mail Server*	mal.google.com
Domain	
CDO host machine	
Enable SSL	□
Port	110
User Id	MailSource@company.com
Password	
Confirm Password	
Search criteria *	

Database Source



Edit Database Source: EvalXform_DBSource

Standard Properties

Name*: EvalXform_DBSource

Description*: Database Source having insurance data

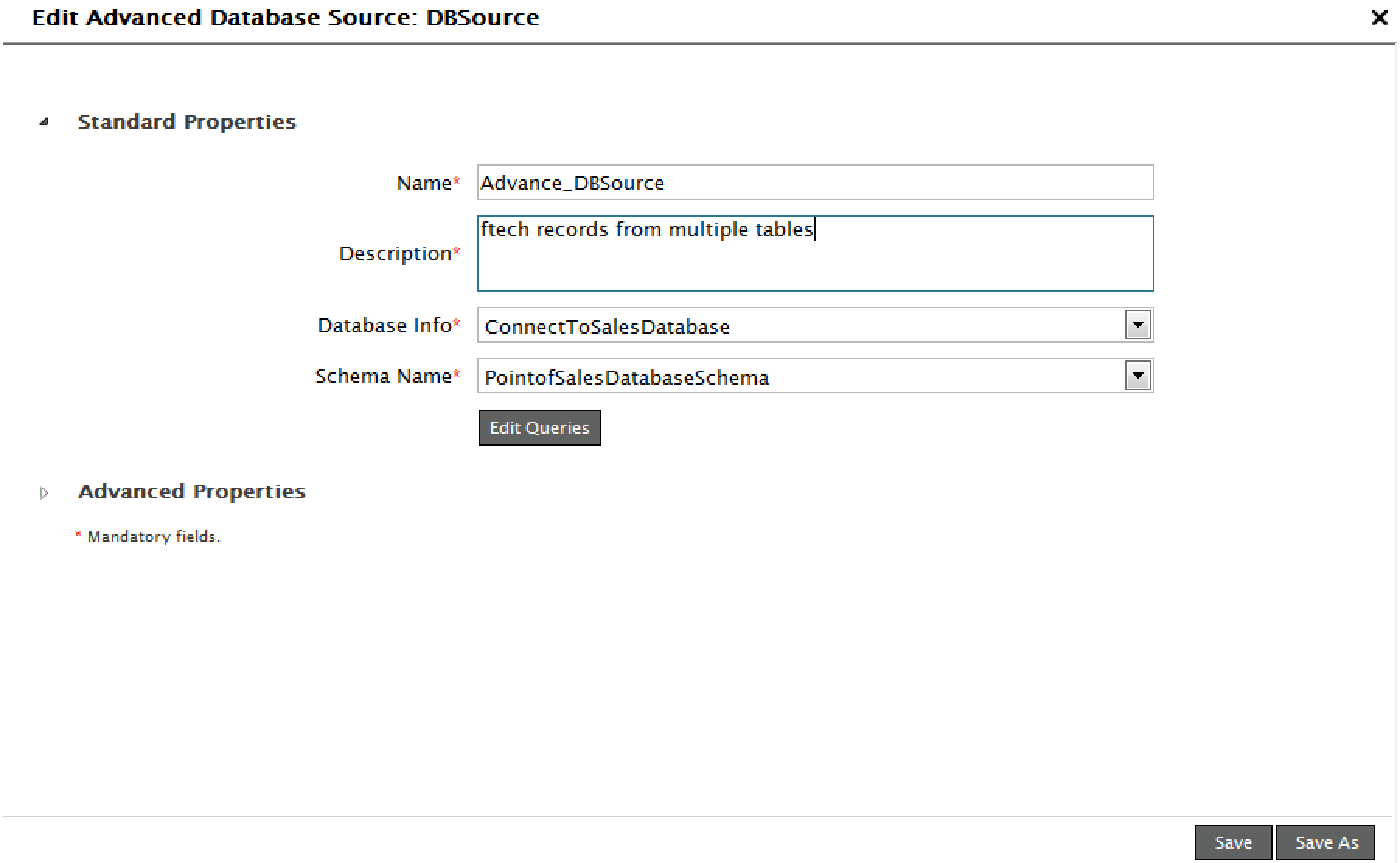
Database Info*: EvalXform_DbInfo

Schema Name*: EvalXform_DbSchema

Advanced Properties

* Mandatory fields.

Advance Database Source

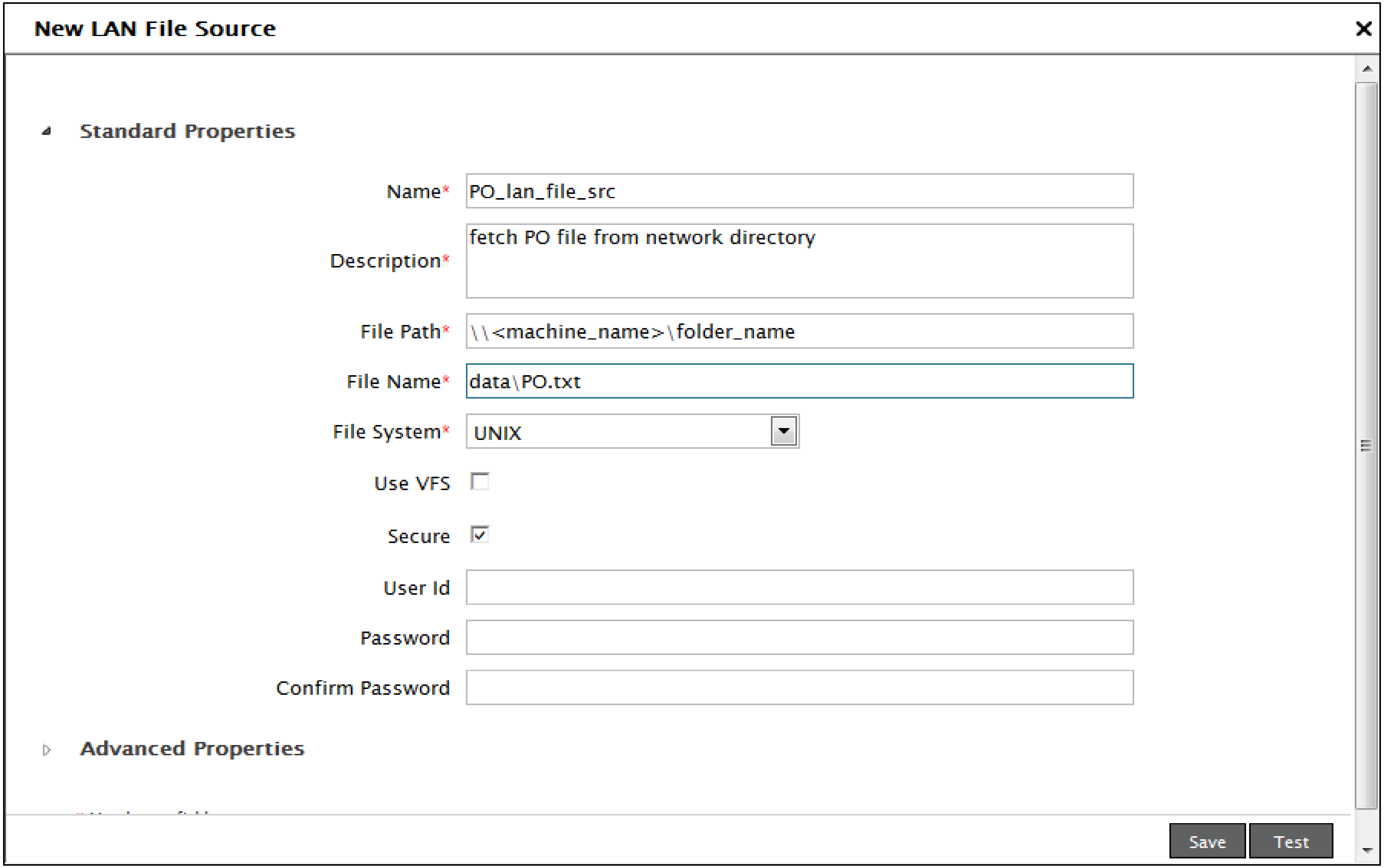


Edit Advanced Database Source: DBSource

Standard Properties

Name*	Advance_DBSource
Description*	ftech records from multiple tables
Database Info*	ConnectToSalesDatabase
Schema Name*	PointofSalesDatabaseSchema

LAN File Source

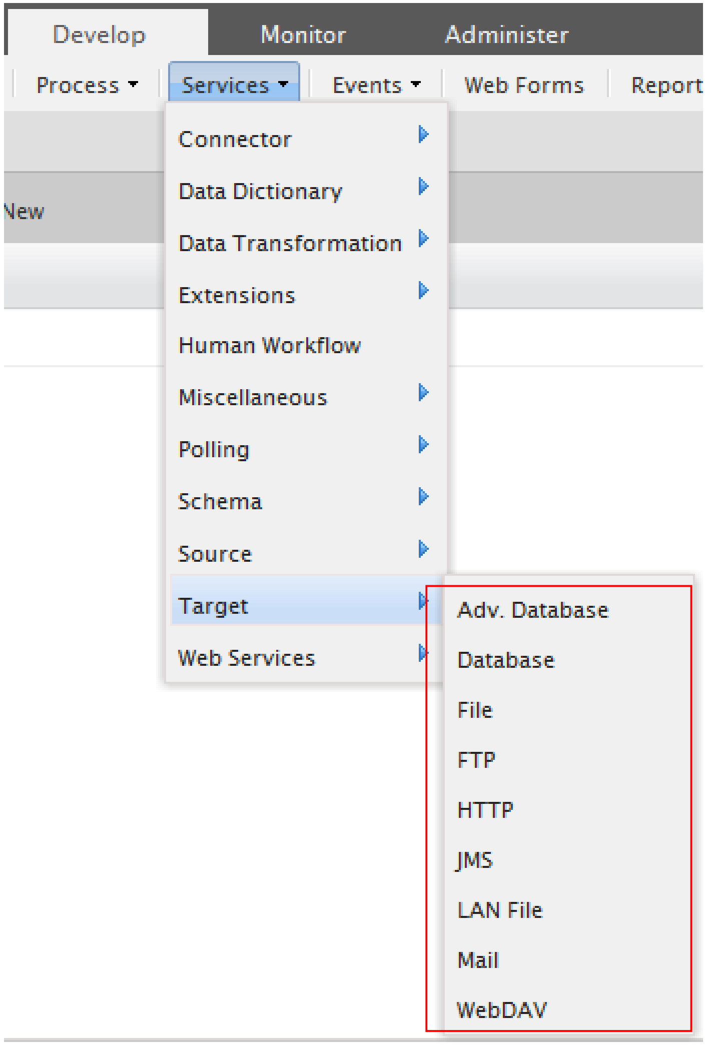


New LAN File Source ✕

Standard Properties

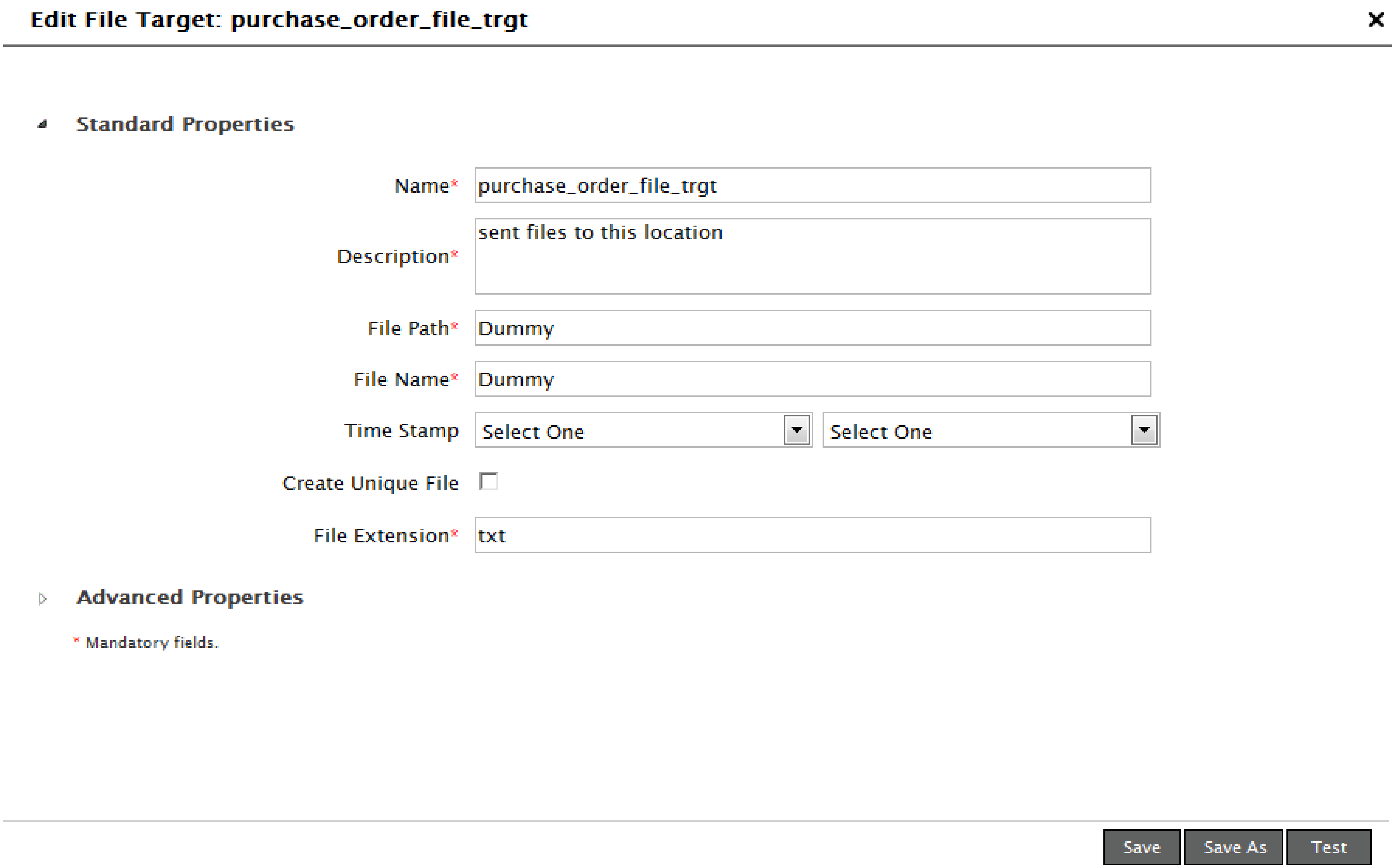
Name*	PO_lan_file_src
Description*	fetch PO file from network directory
File Path*	\\<machine_name>\folder_name
File Name*	data\PO.txt
File System*	UNIX ▾
Use VFS	□
Secure	☑
User Id	
Password	
Confirm Password	

Advanced Properties



Target Services

File Target



Edit File Target: purchase_order_file_trgt

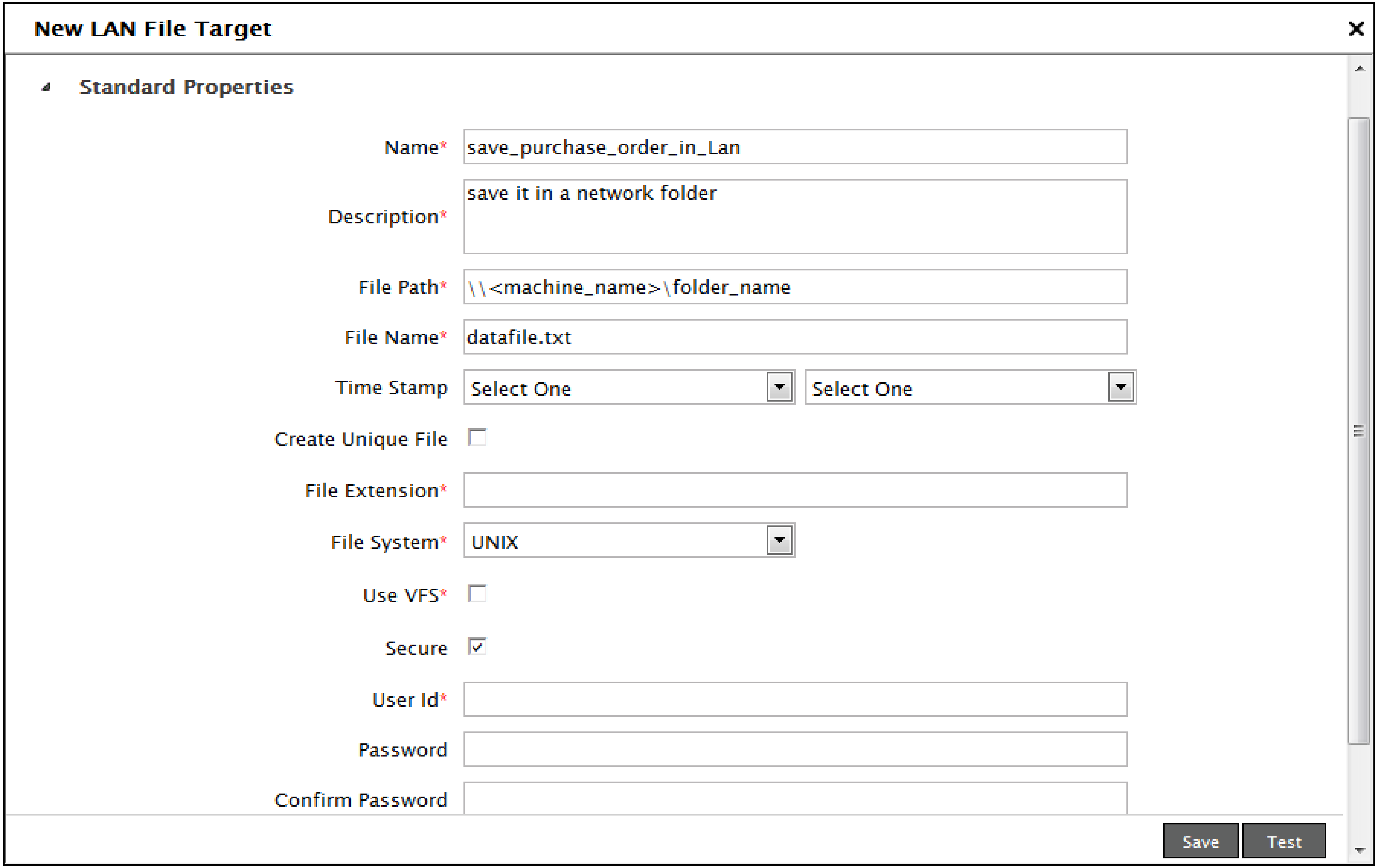
Standard Properties

Name*	purchase_order_file_trgt
Description*	sent files to this location
File Path*	Dummy
File Name*	Dummy
Time Stamp	Select One / Select One
Create Unique File	□
File Extension*	txt

Advanced Properties

* Mandatory fields.

LAN File Target

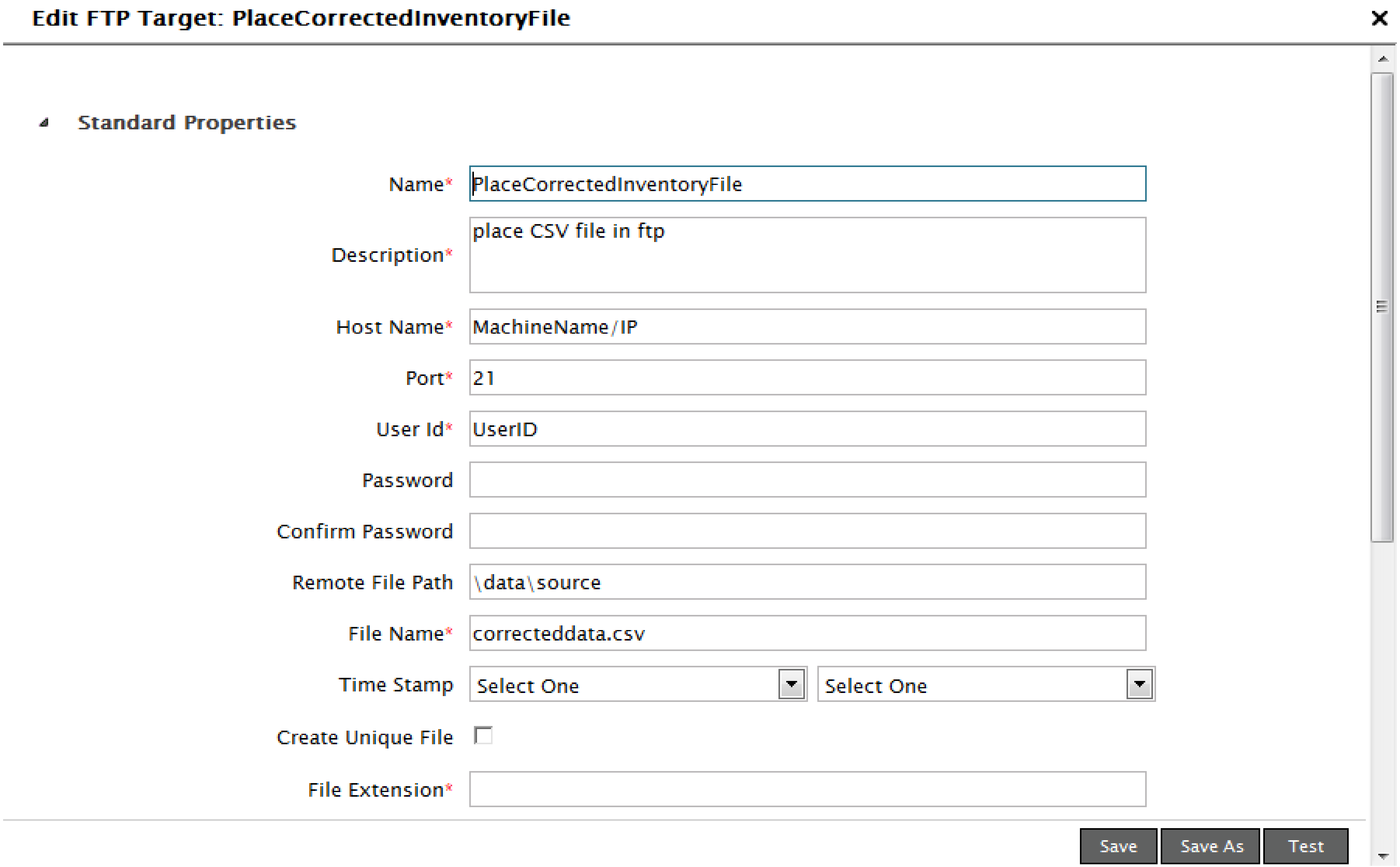


New LAN File Target

Standard Properties

Name*	save_purchase_order_in_Lan
Description*	save it in a network folder
File Path*	\\<machine_name>\folder_name
File Name*	datafile.txt
Time Stamp	Select One / Select One
Create Unique File	□
File Extension*	
File System*	UNIX
Use VFS*	□
Secure	☑
User Id*	
Password	
Confirm Password	

FTP Target

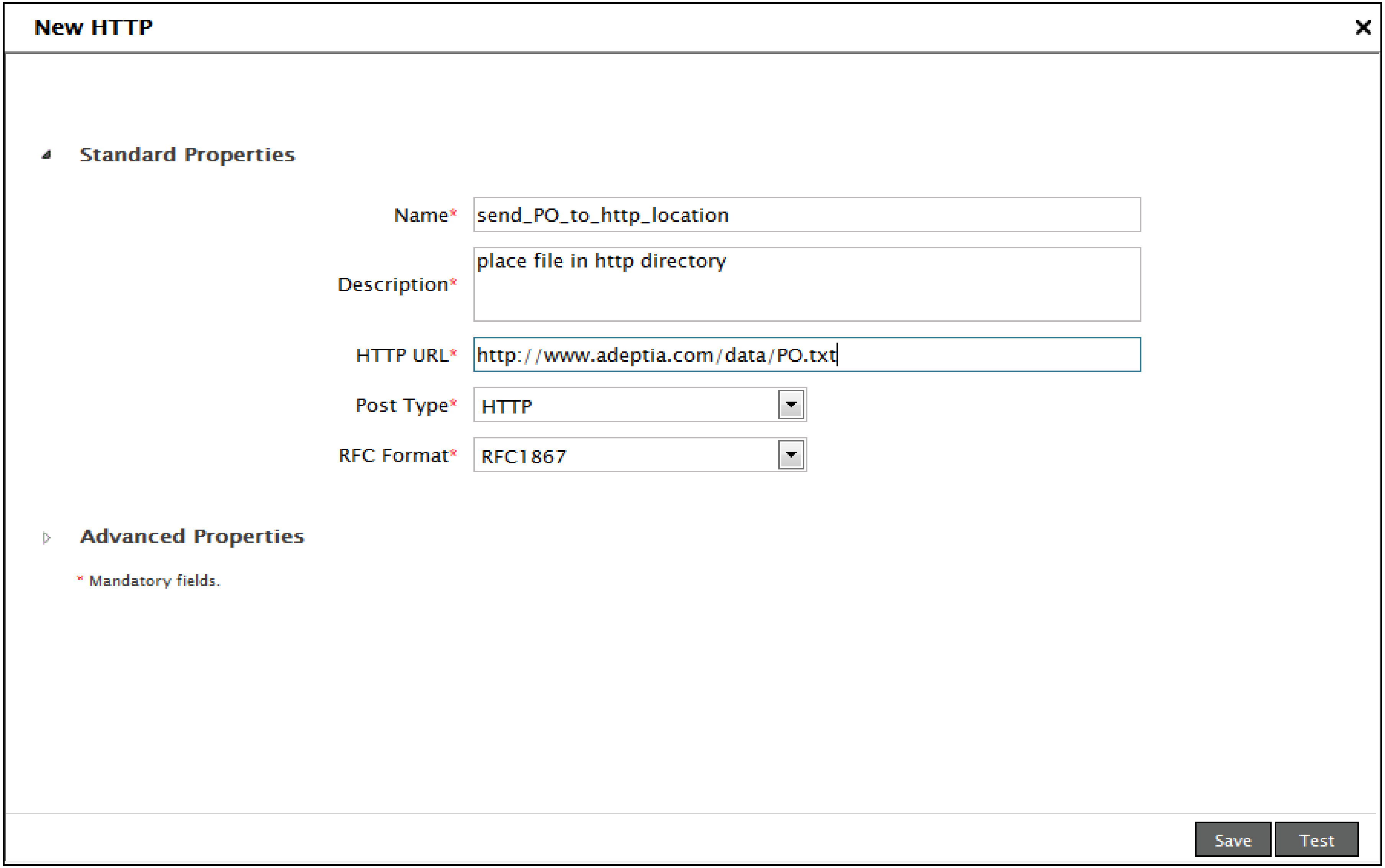


Edit FTP Target: PlaceCorrectedInventoryFile

Standard Properties

Name*	PlaceCorrectedInventoryFile
Description*	place CSV file in ftp
Host Name*	MachineName/IP
Port*	21
User Id*	UserID
Password	
Confirm Password	
Remote File Path	\data\source
File Name*	correcteddata.csv
Time Stamp	Select One Select One
Create Unique File	□
File Extension*	

HTTP Target



New HTTP

Standard Properties

Name* send_PO_to_http_location

Description* place file in http directory

HTTP URL* http://www.adeptia.com/data/PO.txt

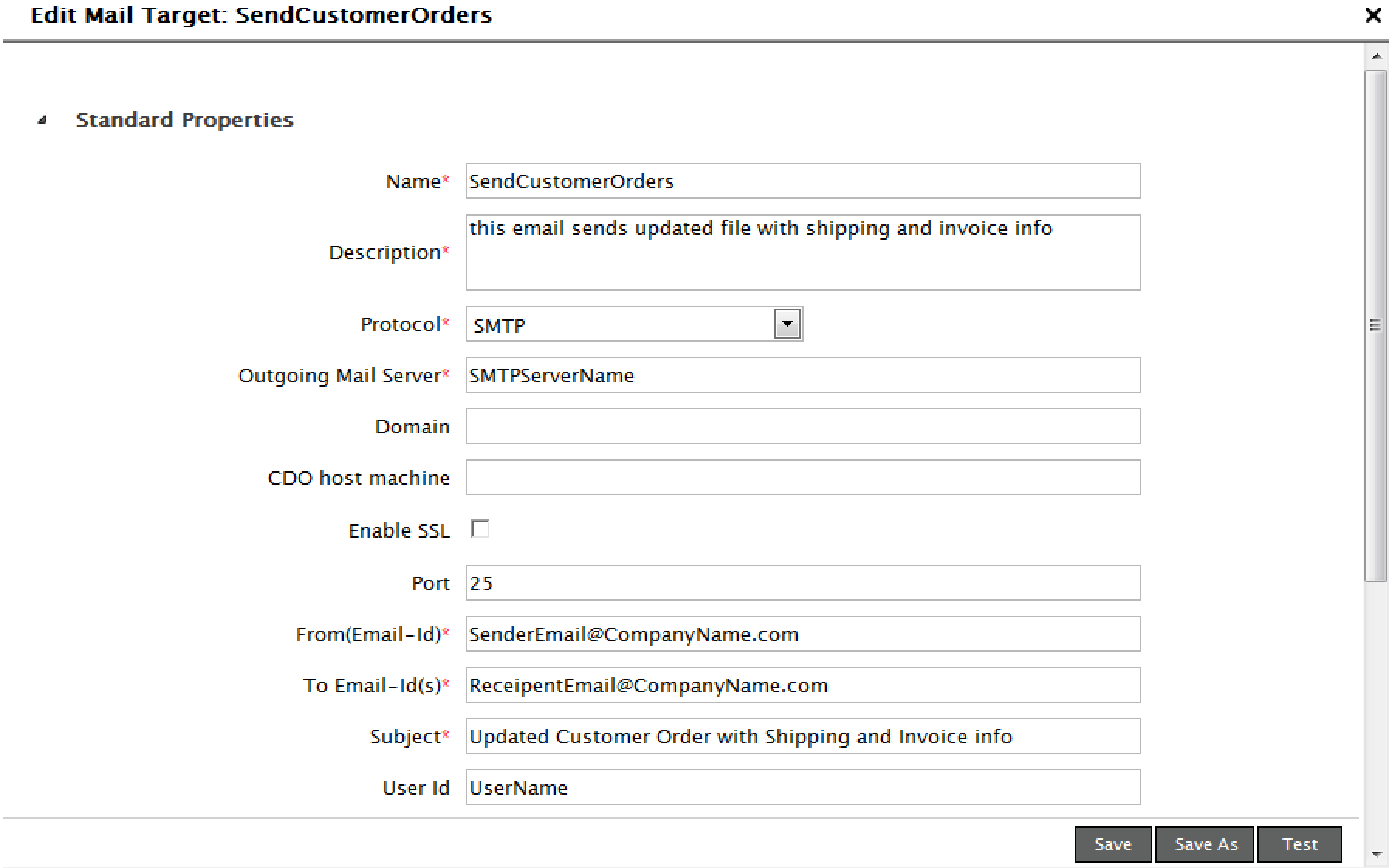
Post Type* HTTP

RFC Format* RFC1867

Advanced Properties

* Mandatory fields.

Mail Target

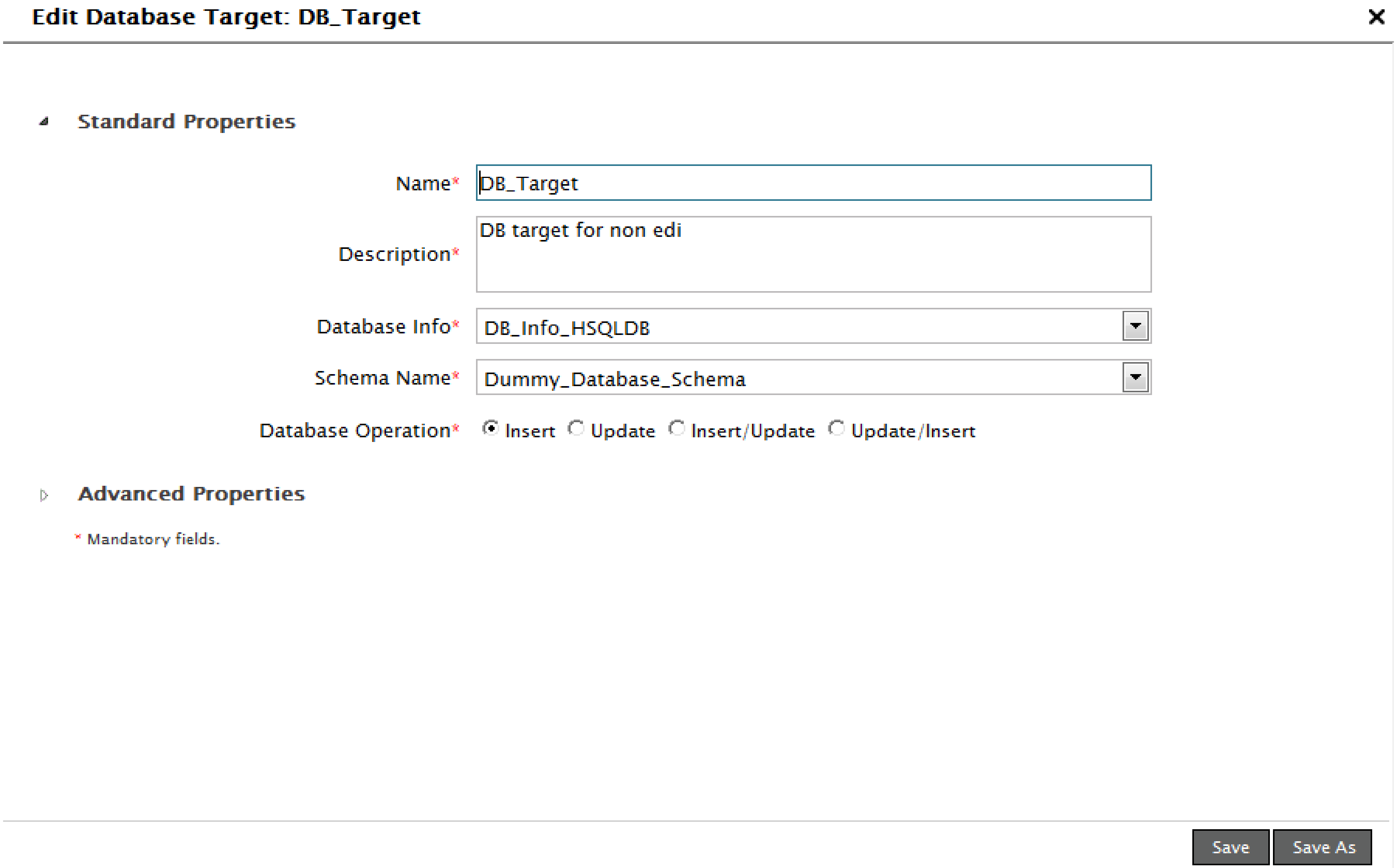


Edit Mail Target: SendCustomerOrders

Standard Properties

Name*	SendCustomerOrders
Description*	this email sends updated file with shipping and invoice info
Protocol*	SMTP
Outgoing Mail Server*	SMTPServerName
Domain	
CDO host machine	
Enable SSL	□
Port	25
From(Email-Id)*	SenderEmail@CompanyName.com
To Email-Id(s)*	ReceipentEmail@CompanyName.com
Subject*	Updated Customer Order with Shipping and Invoice info
User Id	UserName

Database Target



Edit Database Target: DB_Target

Standard Properties

Name* DB_Target

Description* DB target for non edi

Database Info* DB_Info_HSQLDB

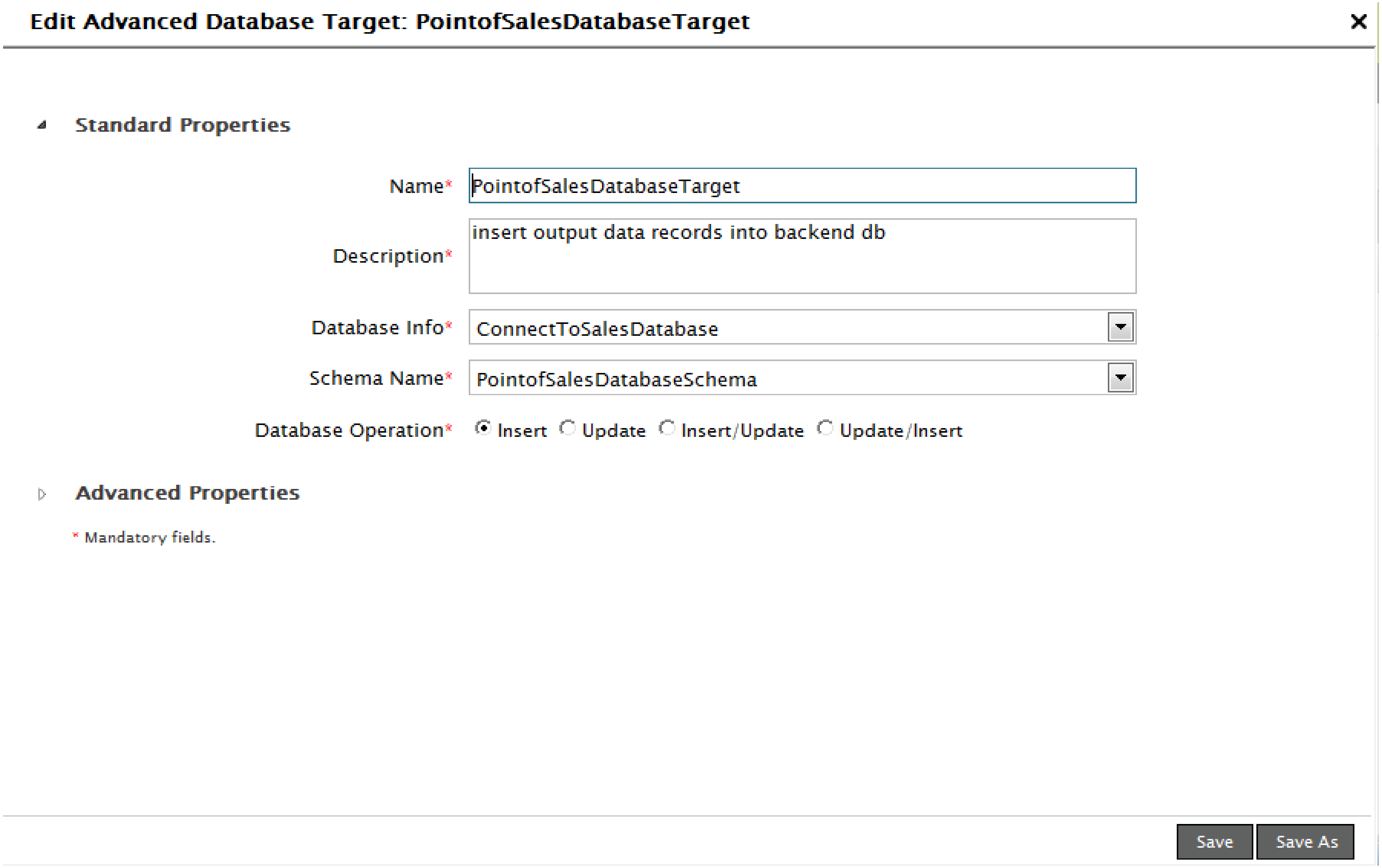
Schema Name* Dummy_Database_Schema

Database Operation* ● Insert ○ Update ○ Insert/Update ○ Update/Insert

Advanced Properties

* Mandatory fields.

Advance Database Target



Edit Advanced Database Target: PointofSalesDatabaseTarget

Standard Properties

Name* PointofSalesDatabaseTarget

Description* insert output data records into backend db

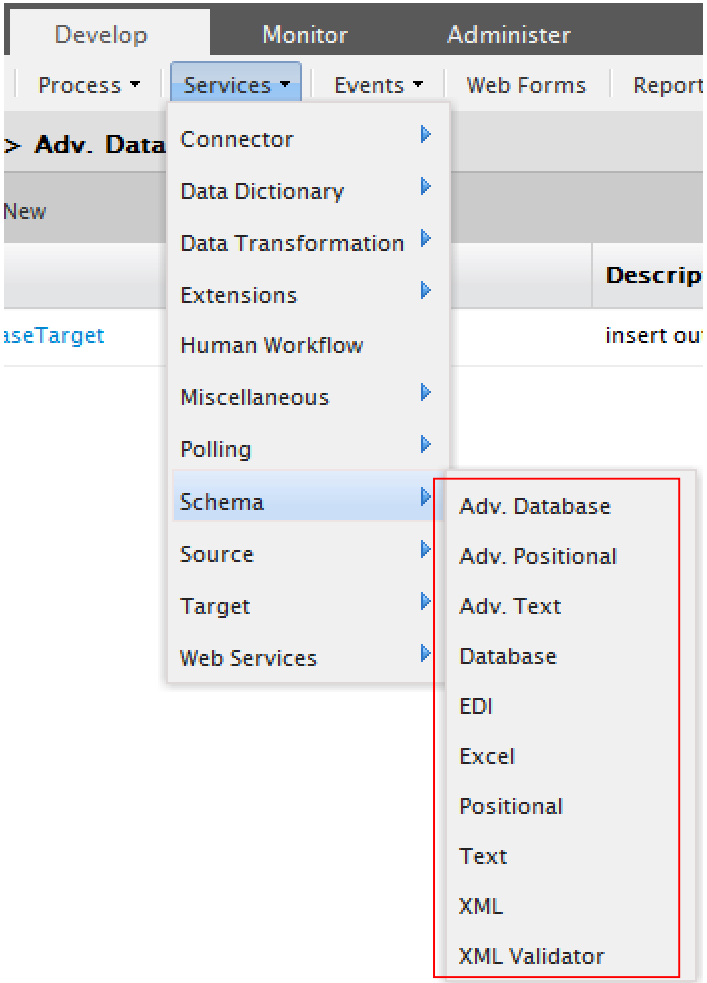
Database Info* ConnectToSalesDatabase

Schema Name* PointofSalesDatabaseSchema

Database Operation* ● Insert ○ Update ○ Insert/Update ○ Update/Insert

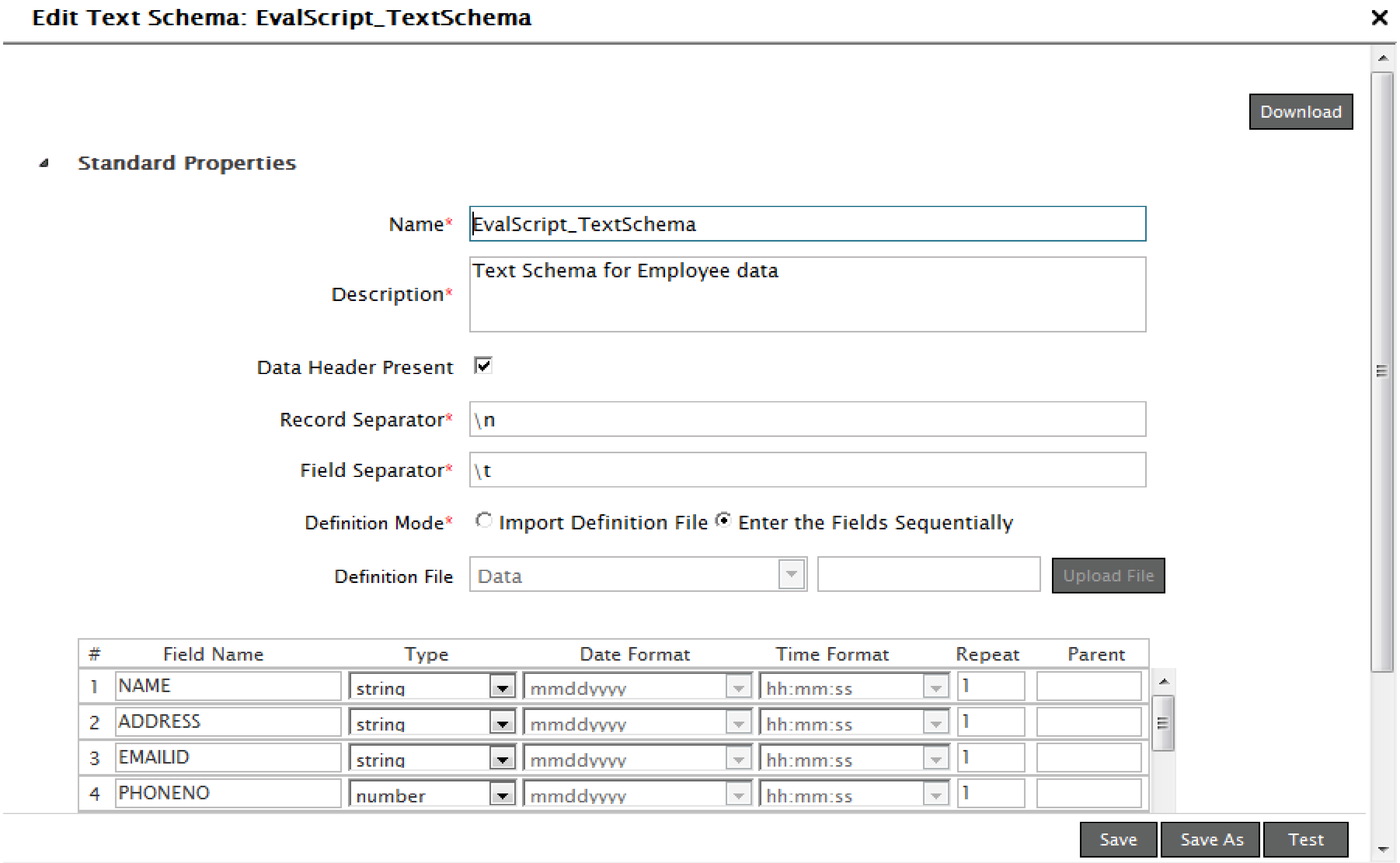
Advanced Properties

* Mandatory fields.



Schema Services

Text Schema



Edit Text Schema: EvalScript_TextSchema

Standard Properties

Name*: EvalScript_TextSchema

Description*: Text Schema for Employee data

Data Header Present: ☑

Record Separator*: \n

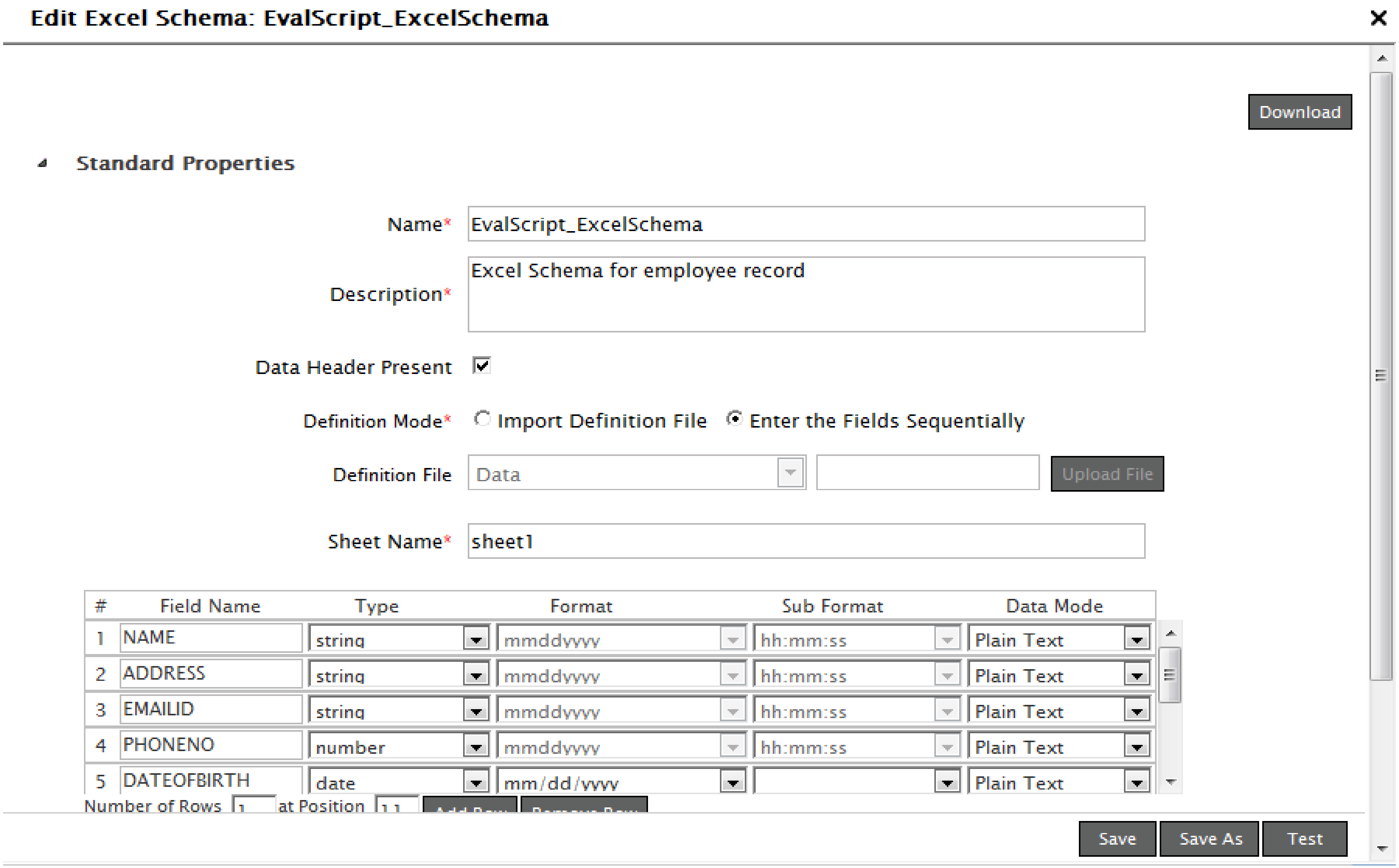
Field Separator*: \t

Definition Mode*: ○ Import Definition File ● Enter the Fields Sequentially

Definition File: Data

#	Field Name	Type	Date Format	Time Format	Repeat	Parent
1	NAME	string	mmddyyyy	hh:mm:ss	1	
2	ADDRESS	string	mmddyyyy	hh:mm:ss	1	
3	EMAILID	string	mmddyyyy	hh:mm:ss	1	
4	PHONENO	number	mmddyyyy	hh:mm:ss	1	

Excel Schema



Edit Excel Schema: EvalScript_ExcelSchema

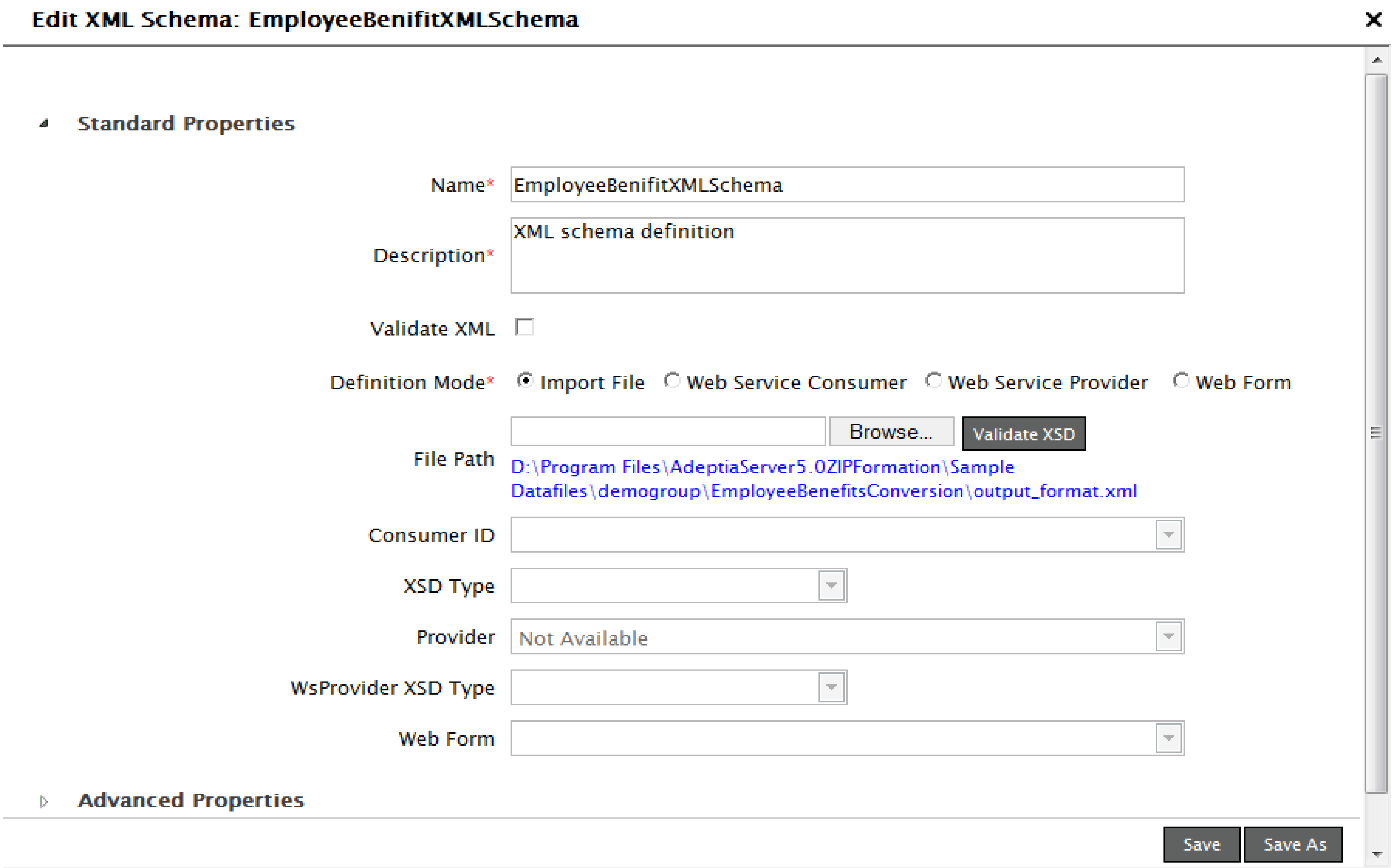
Standard Properties

Name*	EvalScript_ExcelSchema
Description*	Excel Schema for employee record
Data Header Present	☑
Definition Mode*	○ Import Definition File ● Enter the Fields Sequentially
Definition File	Data
Sheet Name*	sheet1

#	Field Name	Type	Format	Sub Format	Data Mode
1	NAME	string	mmddyyyy	hh:mm:ss	Plain Text
2	ADDRESS	string	mmddyyyy	hh:mm:ss	Plain Text
3	EMAILID	string	mmddyyyy	hh:mm:ss	Plain Text
4	PHONENO	number	mmddyyyy	hh:mm:ss	Plain Text
5	DATEOFBIRTH	date	mm/dd/yyyy		Plain Text

Number of Rows 1 at Position 11 Add Row Remove Row

XML Schema



Edit XML Schema: EmployeeBenifitXMLSchema

Standard Properties

Name* EmployeeBenifitXMLSchema

Description* XML schema definition

Validate XML □

Definition Mode* ● Import File ○ Web Service Consumer ○ Web Service Provider ○ Web Form

File Path D:\Program Files\AdeptiaServer5.0ZIPFormation\Sample Datafiles\demogroup\EmployeeBenefitsConversion\output_format.xml

Consumer ID

XSD Type

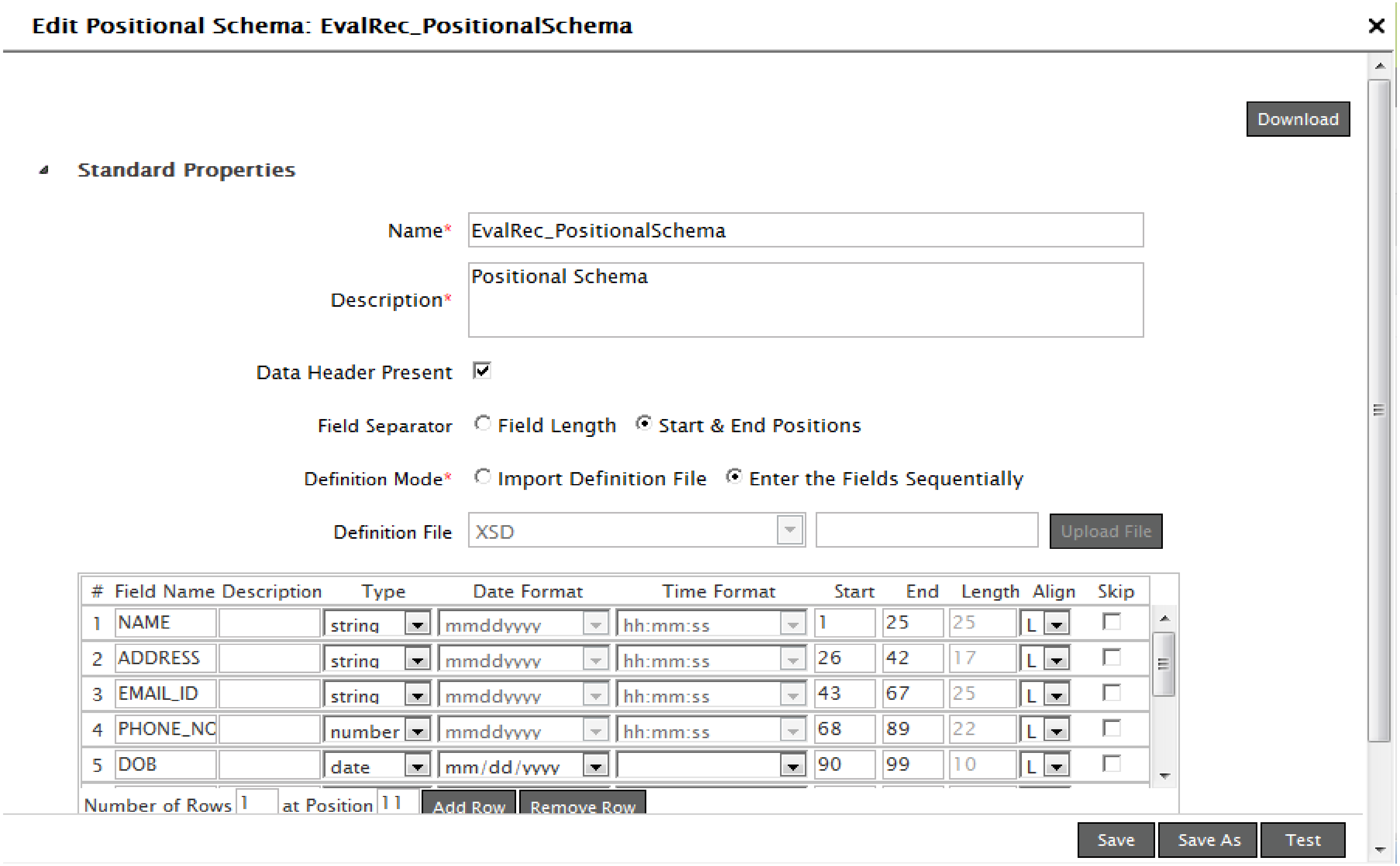
Provider Not Available

WsProvider XSD Type

Web Form

Advanced Properties

Positional Schema



Edit Positional Schema: EvalRec_PositionalSchema

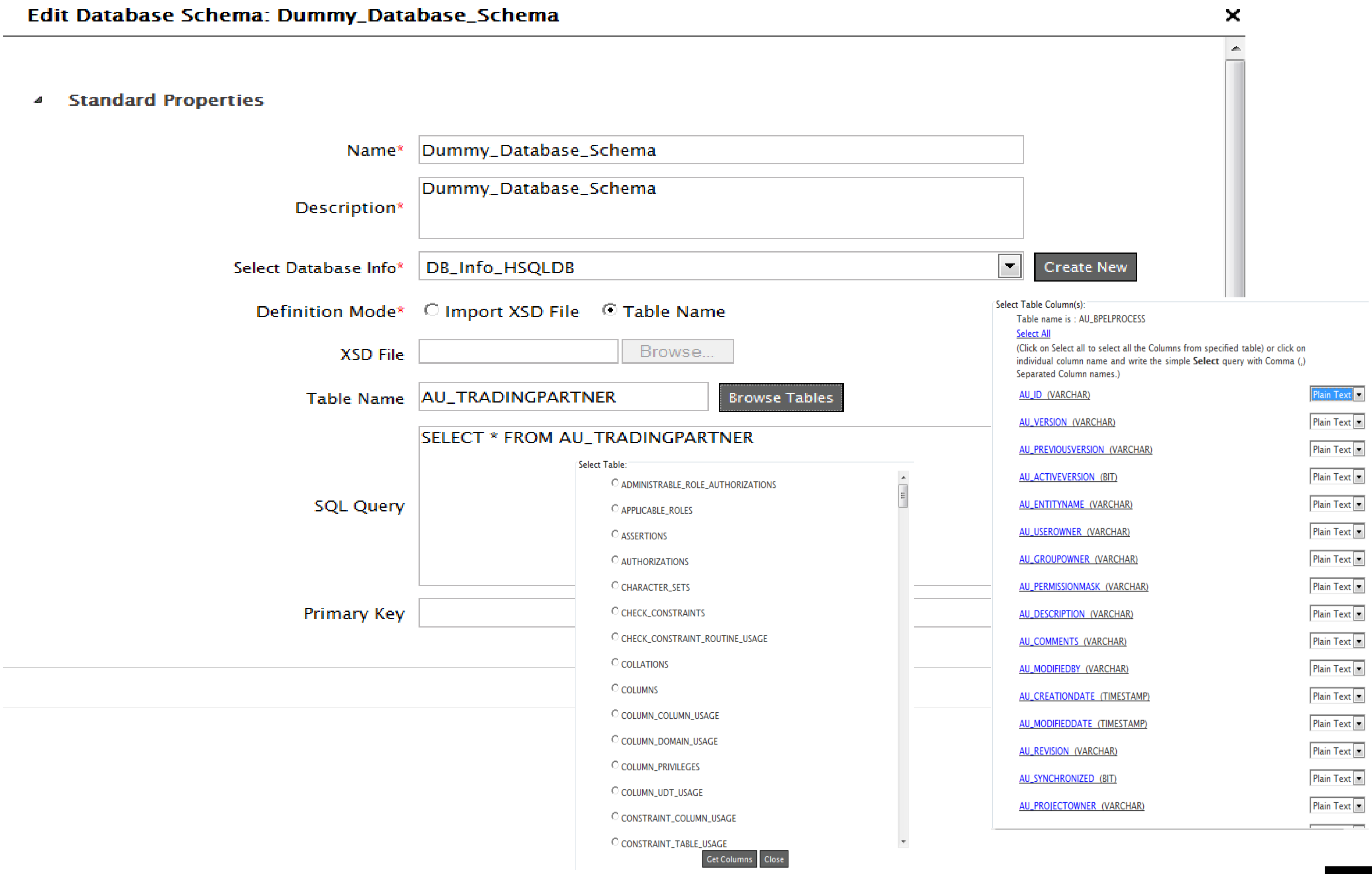
Standard Properties

Name*	EvalRec_PositionalSchema
Description*	Positional Schema
Data Header Present	☑
Field Separator	○ Field Length ● Start & End Positions
Definition Mode*	○ Import Definition File ● Enter the Fields Sequentially
Definition File	XSD

#	Field Name	Description	Type	Date Format	Time Format	Start	End	Length	Align	Skip
1	NAME		string	mmddyyyy	hh:mm:ss	1	25	25	L	□
2	ADDRESS		string	mmddyyyy	hh:mm:ss	26	42	17	L	□
3	EMAIL_ID		string	mmddyyyy	hh:mm:ss	43	67	25	L	□
4	PHONE_NC		number	mmddyyyy	hh:mm:ss	68	89	22	L	□
5	DOB		date	mm/dd/yyyy		90	99	10	L	□

Number of Rows 1 at Position 11

Database Schema



Edit Database Schema: Dummy_Database_Schema

Standard Properties

Name* Dummy_Database_Schema

Description* Dummy_Database_Schema

Select Database Info* DB_Info_HSQLDB

Definition Mode* ○ Import XSD File ● Table Name

XSD File

Table Name AU_TRADINGPARTNER

SQL Query SELECT * FROM AU_TRADINGPARTNER

Primary Key

Select Table:

- ○ ADMINISTRABLE_ROLE_AUTHORIZATIONS
- ○ APPLICABLE_ROLES
- ○ ASSERTIONS
- ○ AUTHORIZATIONS
- ○ CHARACTER_SETS
- ○ CHECK_CONSTRAINTS
- ○ CHECK_CONSTRAINT_ROUTINE_USAGE
- ○ COLLATIONS
- ○ COLUMNS
- ○ COLUMN_COLUMN_USAGE
- ○ COLUMN_DOMAIN_USAGE
- ○ COLUMN_PRIVILEGES
- ○ COLUMN_UDT_USAGE
- ○ CONSTRAINT_COLUMN_USAGE
- ○ CONSTRAINT_TABLE_USAGE

Select Table Column(s):

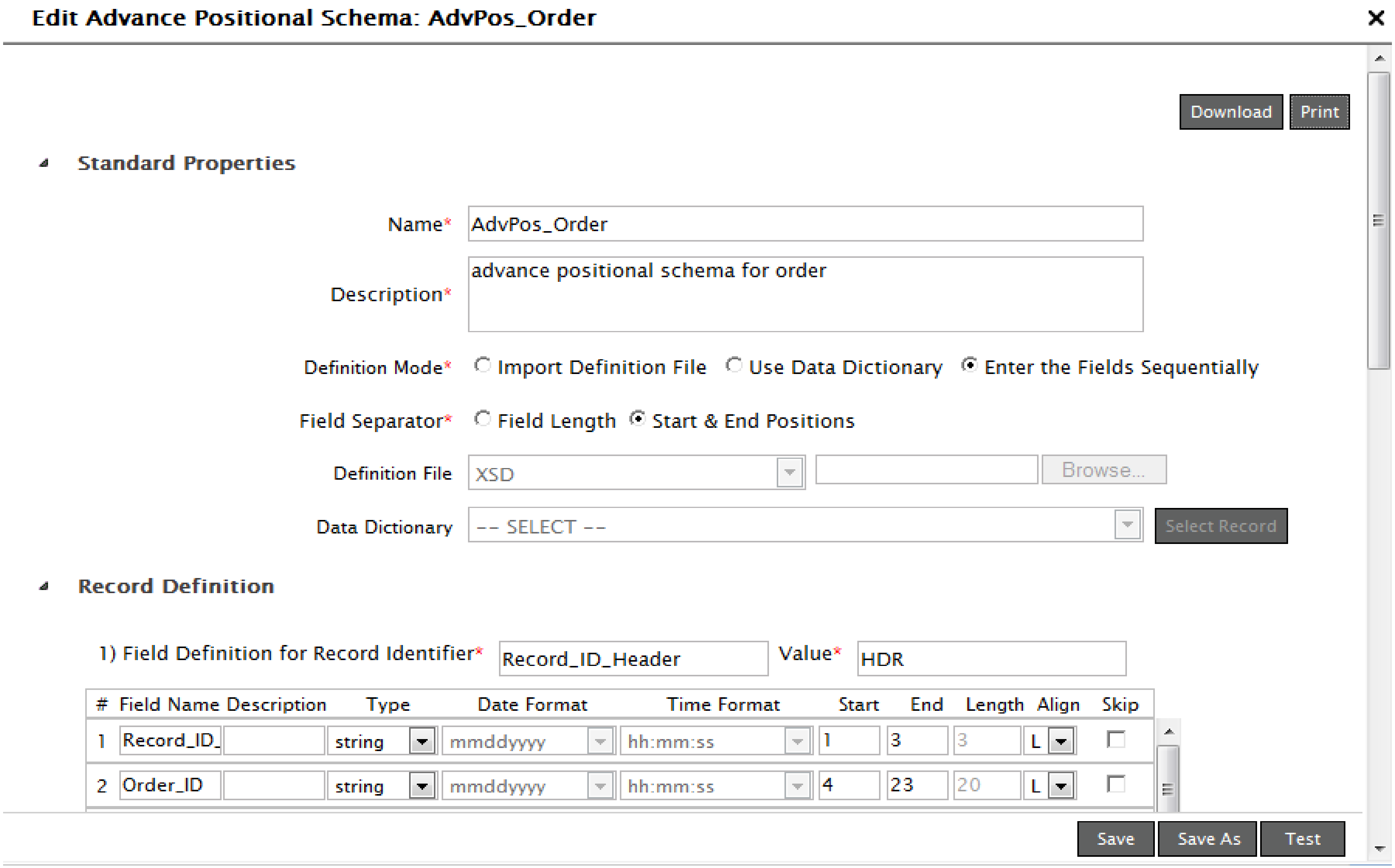
Table name is : AU_BPELPROCESS

Select All

(Click on Select all to select all the Columns from specified table) or click on individual column name and write the simple **Select** query with Comma (,) Separated Column names.)

AU_ID (VARCHAR)	Plain Text
AU_VERSION (VARCHAR)	Plain Text
AU_PREVIOUSVERSION (VARCHAR)	Plain Text
AU_ACTIVEVERSION (BIT)	Plain Text
AU_ENTITYNAME (VARCHAR)	Plain Text
AU_USEROWNER (VARCHAR)	Plain Text
AU_GROUPOWNER (VARCHAR)	Plain Text
AU_PERMISSIONMASK (VARCHAR)	Plain Text
AU_DESCRIPTION (VARCHAR)	Plain Text
AU_COMMENTS (VARCHAR)	Plain Text
AU_MODIFIEDBY (VARCHAR)	Plain Text
AU_CREATIONDATE (TIMESTAMP)	Plain Text
AU_MODIFIEDDATE (TIMESTAMP)	Plain Text
AU_REVISION (VARCHAR)	Plain Text
AU_SYNCHRONIZED (BIT)	Plain Text
AU_PROJECTOWNER (VARCHAR)	Plain Text

Advance Positional Schema



Edit Advance Positional Schema: AdvPos_Order

Standard Properties

Name*: AdvPos_Order

Description*: advance positional schema for order

Definition Mode*: ○ Import Definition File ○ Use Data Dictionary ● Enter the Fields Sequentially

Field Separator*: ○ Field Length ● Start & End Positions

Definition File: XSD

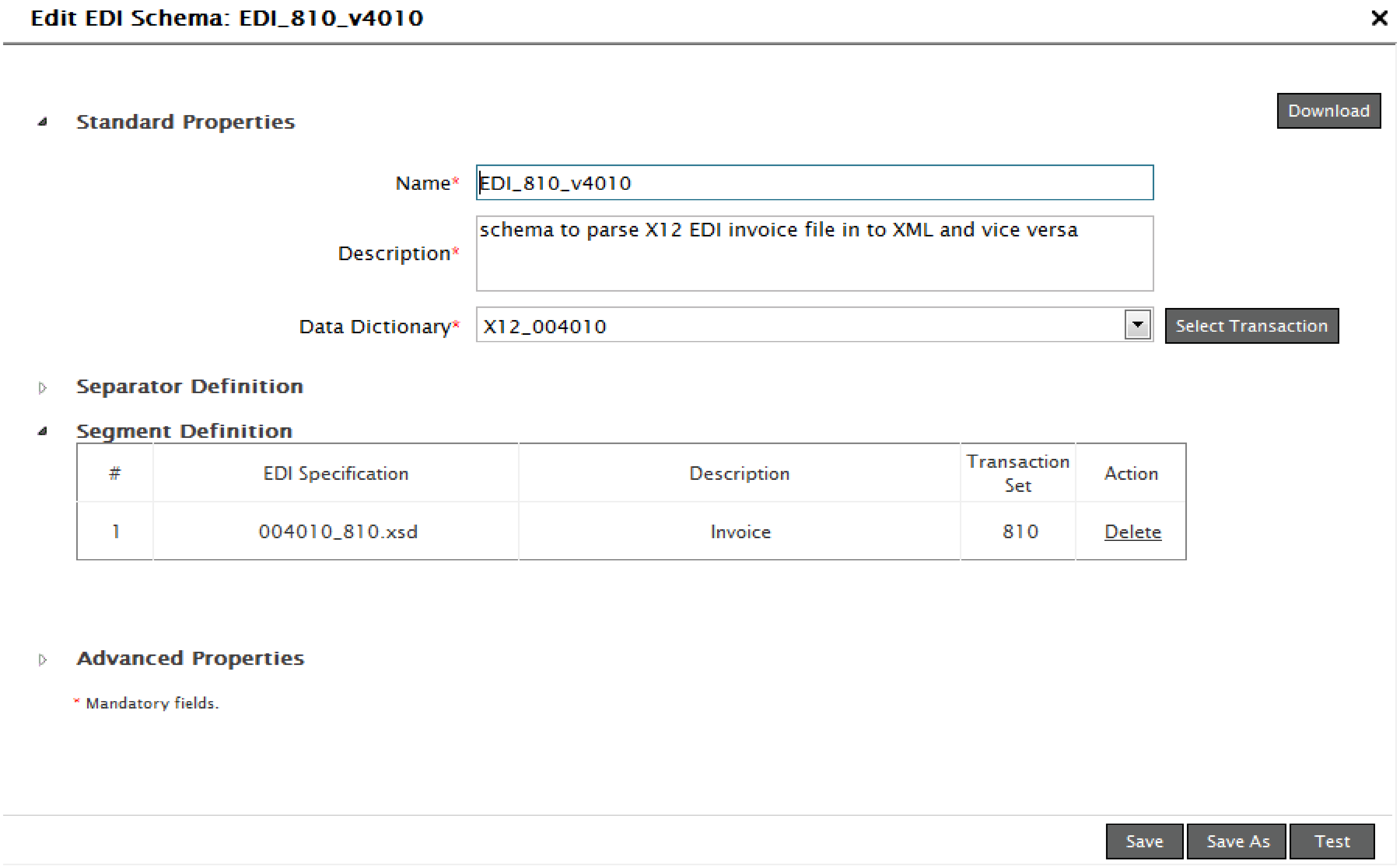
Data Dictionary: -- SELECT --

Record Definition

1) Field Definition for Record Identifier*: Record_ID_Header Value*: HDR

#	Field Name	Description	Type	Date Format	Time Format	Start	End	Length	Align	Skip
1	Record_ID_		string	mmddyyyy	hh:mm:ss	1	3	3	L	□
2	Order_ID		string	mmddyyyy	hh:mm:ss	4	23	20	L	□

EDI Schema



Edit EDI Schema: EDI_810_v4010

Standard Properties

Name* EDI_810_v4010

Description* schema to parse X12 EDI invoice file in to XML and vice versa

Data Dictionary* X12_004010

Separator Definition

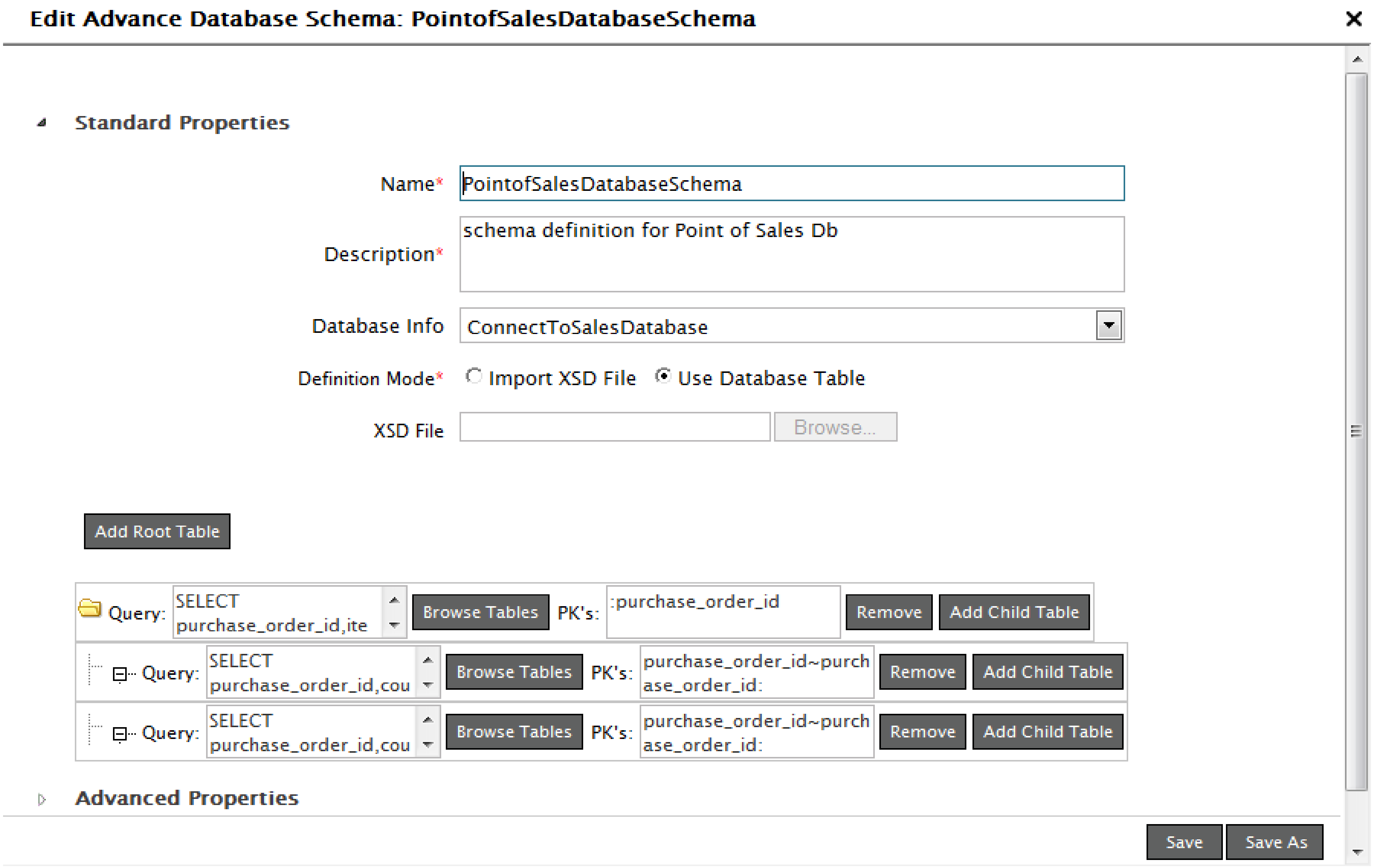
Segment Definition

#	EDI Specification	Description	Transaction Set	Action
1	004010_810.xsd	Invoice	810	Delete

Advanced Properties

* Mandatory fields.

Advance Database Schema (Step 1)



Edit Advance Database Schema: PointofSalesDatabaseSchema

Standard Properties

Name*: PointofSalesDatabaseSchema

Description*: schema definition for Point of Sales Db

Database Info: ConnectToSalesDatabase

Definition Mode*: ○ Import XSD File ● Use Database Table

XSD File: Browse...

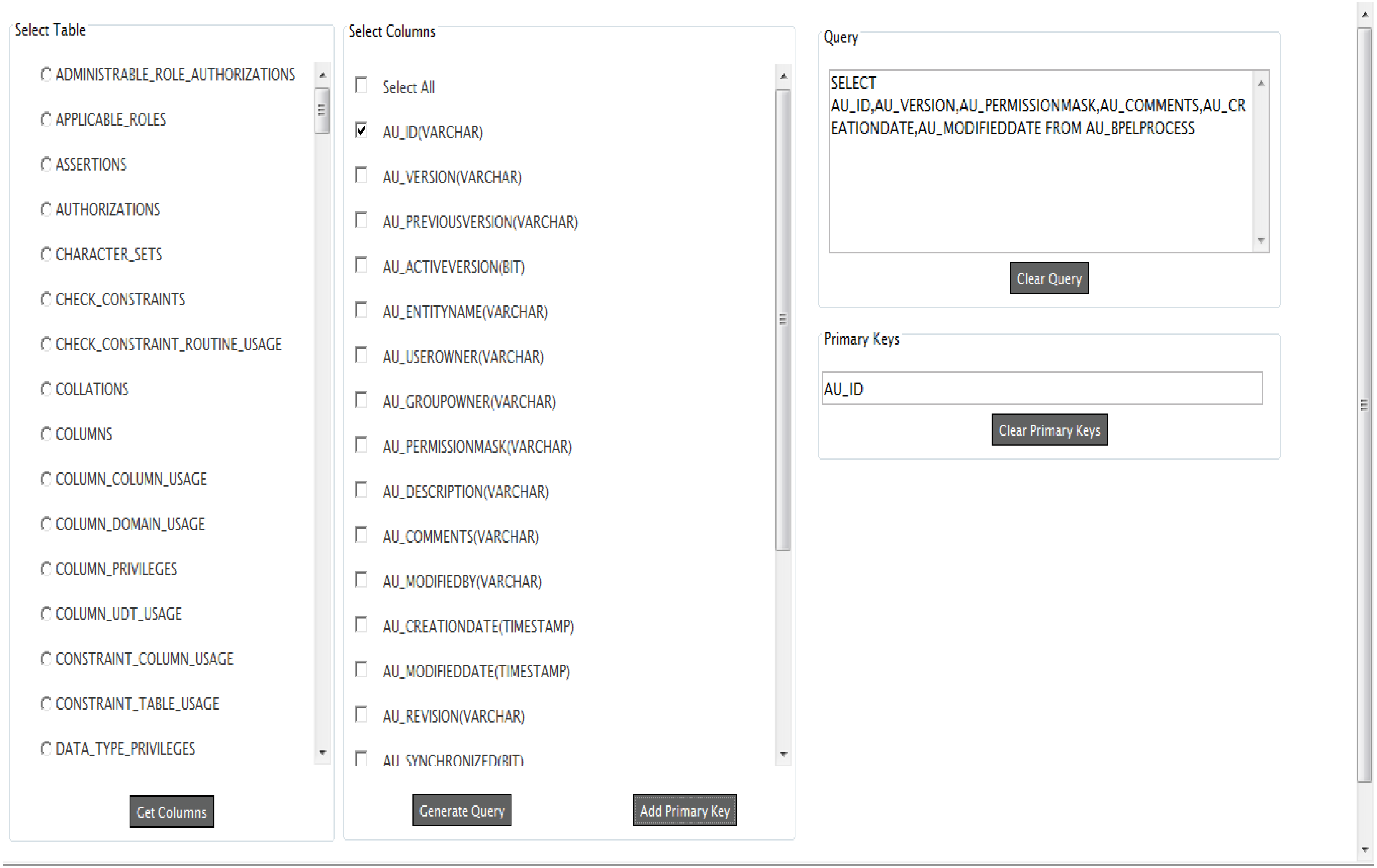
Query: SELECT purchase_order_id,ite — Browse Tables — PK's: :purchase_order_id — Remove — Add Child Table

Query: SELECT purchase_order_id,cou — Browse Tables — PK's: purchase_order_id~purchase_order_id: — Remove — Add Child Table

Query: SELECT purchase_order_id,cou — Browse Tables — PK's: purchase_order_id~purchase_order_id: — Remove — Add Child Table

Advanced Properties

Advance Database Schema (Step 2)



Select Table

○ ADMINISTRABLE_ROLE_AUTHORIZATIONS
○ APPLICABLE_ROLES
○ ASSERTIONS
○ AUTHORIZATIONS
○ CHARACTER_SETS
○ CHECK_CONSTRAINTS
○ CHECK_CONSTRAINT_ROUTINE_USAGE
○ COLLATIONS
○ COLUMNS
○ COLUMN_COLUMN_USAGE
○ COLUMN_DOMAIN_USAGE
○ COLUMN_PRIVILEGES
○ COLUMN_UDT_USAGE
○ CONSTRAINT_COLUMN_USAGE
○ CONSTRAINT_TABLE_USAGE
○ DATA_TYPE_PRIVILEGES

Select Columns

□ Select All
☑ AU_ID(VARCHAR)
□ AU_VERSION(VARCHAR)
□ AU_PREVIOUSVERSION(VARCHAR)
□ AU_ACTIVEVERSION(BIT)
□ AU_ENTITYNAME(VARCHAR)
□ AU_USEROWNER(VARCHAR)
□ AU_GROUPOWNER(VARCHAR)
□ AU_PERMISSIONMASK(VARCHAR)
□ AU_DESCRIPTION(VARCHAR)
□ AU_COMMENTS(VARCHAR)
□ AU_MODIFIEDBY(VARCHAR)
□ AU_CREATIONDATE(TIMESTAMP)
□ AU_MODIFIEDDATE(TIMESTAMP)
□ AU_REVISION(VARCHAR)
□ AU_SYNCHRONIZED(BIT)

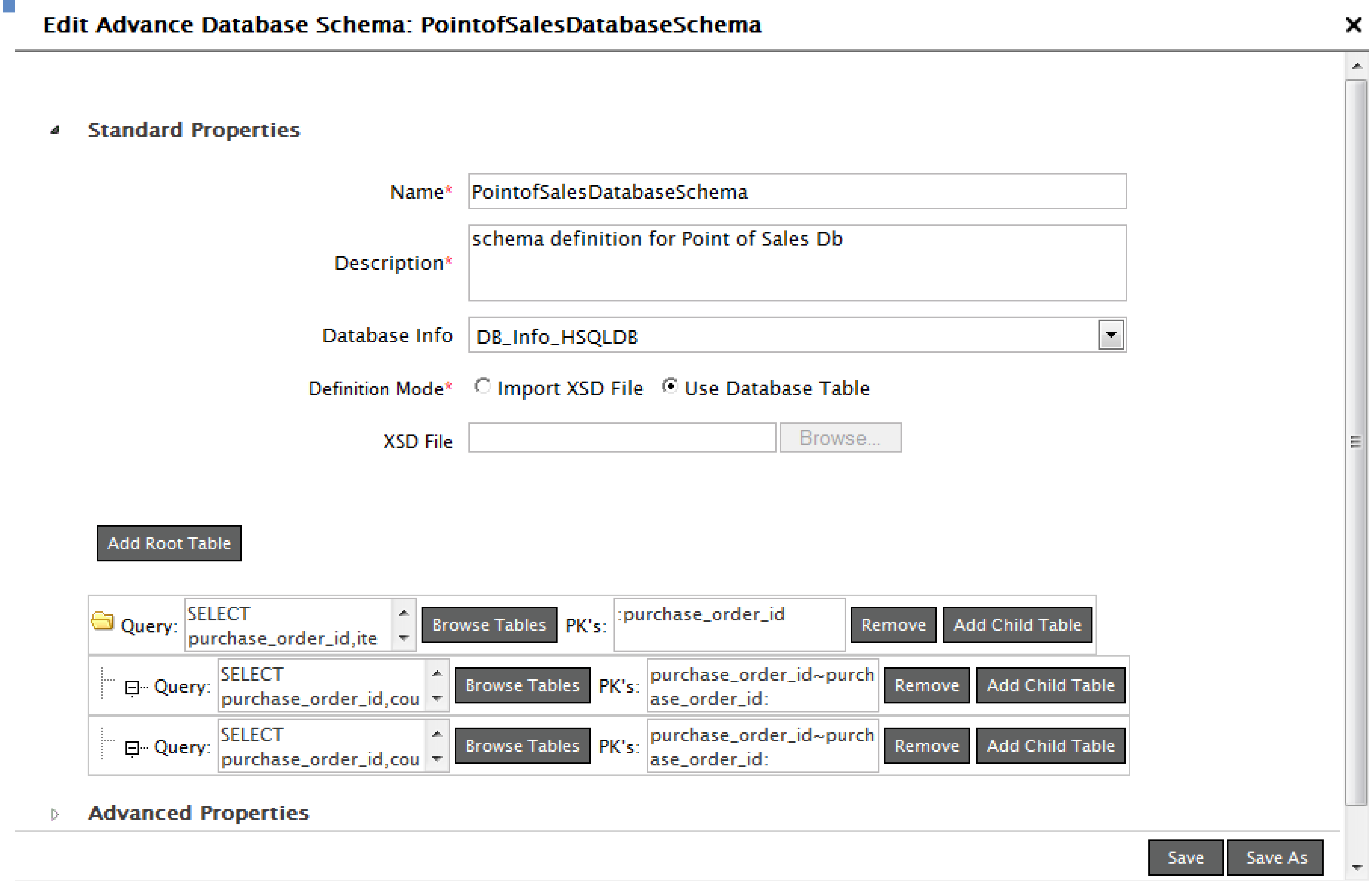
Query

```
SELECT
AU_ID,AU_VERSION,AU_PERMISSIONMASK,AU_COMMENTS,AU_CR
EATIONDATE,AU_MODIFIEDDATE FROM AU_BPELPROCESS
```

Primary Keys

AU_ID

Advance Database Schema (Step 2)



Edit Advance Database Schema: PointofSalesDatabaseSchema ×

Standard Properties

Name*: PointofSalesDatabaseSchema

Description*: schema definition for Point of Sales Db

Database Info: DB_Info_HSQLDB

Definition Mode*: ○ Import XSD File ● Use Database Table

XSD File: Browse...

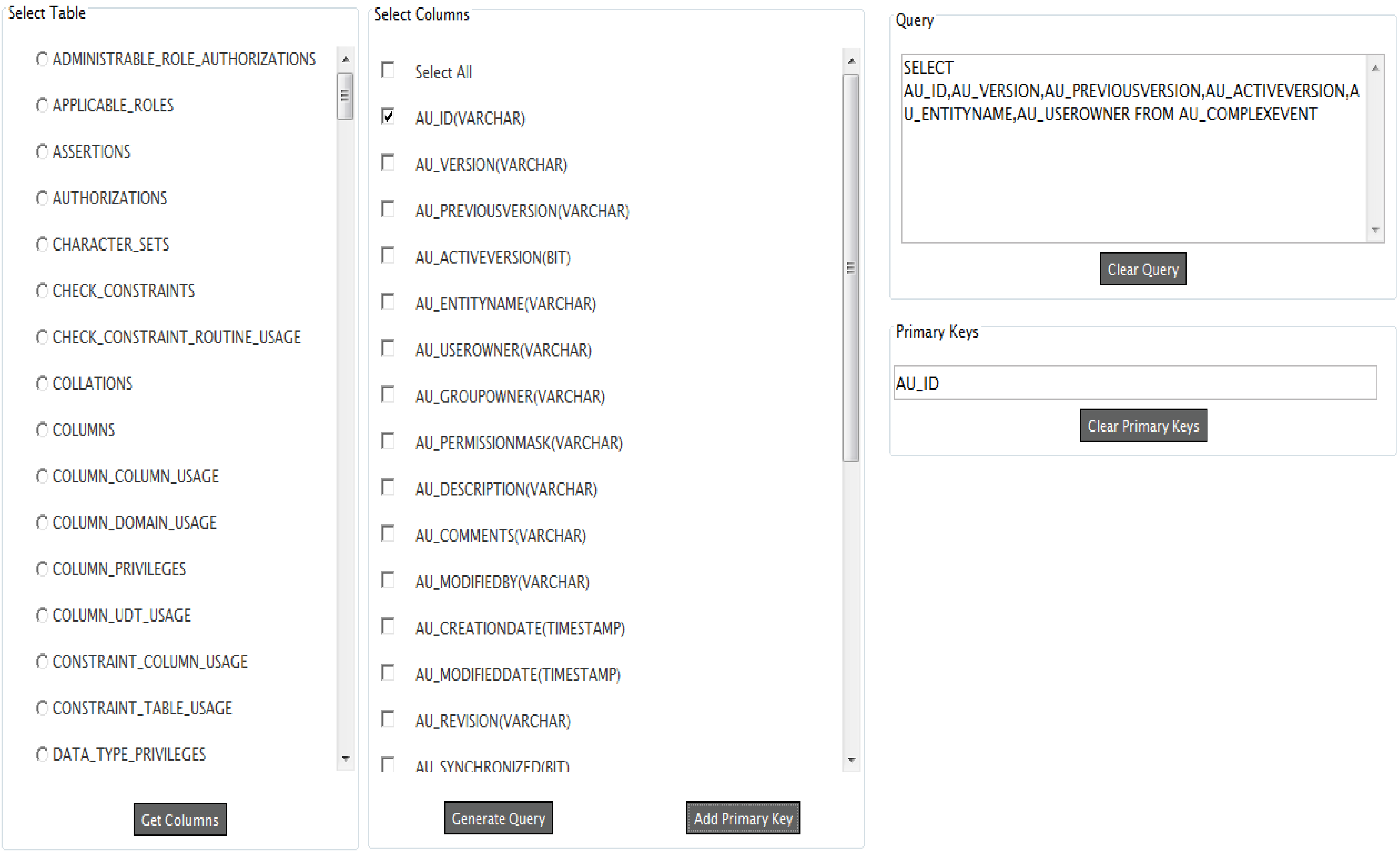
Add Root Table

Query:	SELECT purchase_order_id,ite	Browse Tables	PK's:	:purchase_order_id	Remove / Add Child Table
⊟ Query:	SELECT purchase_order_id,cou	Browse Tables	PK's:	purchase_order_id~purchase_order_id:	Remove / Add Child Table
⊟ Query:	SELECT purchase_order_id,cou	Browse Tables	PK's:	purchase_order_id~purchase_order_id:	Remove / Add Child Table

Advanced Properties

Save | Save As

Advance Database Schema (Step 4)



Select Table

○ ADMINISTRABLE_ROLE_AUTHORIZATIONS
○ APPLICABLE_ROLES
○ ASSERTIONS
○ AUTHORIZATIONS
○ CHARACTER_SETS
○ CHECK_CONSTRAINTS
○ CHECK_CONSTRAINT_ROUTINE_USAGE
○ COLLATIONS
○ COLUMNS
○ COLUMN_COLUMN_USAGE
○ COLUMN_DOMAIN_USAGE
○ COLUMN_PRIVILEGES
○ COLUMN_UDT_USAGE
○ CONSTRAINT_COLUMN_USAGE
○ CONSTRAINT_TABLE_USAGE
○ DATA_TYPE_PRIVILEGES

Get Columns

Select Columns

□ Select All
☑ AU_ID(VARCHAR)
□ AU_VERSION(VARCHAR)
□ AU_PREVIOUSVERSION(VARCHAR)
□ AU_ACTIVEVERSION(BIT)
□ AU_ENTITYNAME(VARCHAR)
□ AU_USEROWNER(VARCHAR)
□ AU_GROUPOWNER(VARCHAR)
□ AU_PERMISSIONMASK(VARCHAR)
□ AU_DESCRIPTION(VARCHAR)
□ AU_COMMENTS(VARCHAR)
□ AU_MODIFIEDBY(VARCHAR)
□ AU_CREATIONDATE(TIMESTAMP)
□ AU_MODIFIEDDATE(TIMESTAMP)
□ AU_REVISION(VARCHAR)
□ AU_SYNCHRONIZED(BIT)

Generate Query   Add Primary Key

Query

```
SELECT
AU_ID,AU_VERSION,AU_PREVIOUSVERSION,AU_ACTIVEVERSION,A
U_ENTITYNAME,AU_USEROWNER FROM AU_COMPLEXEVENT
```

Clear Query

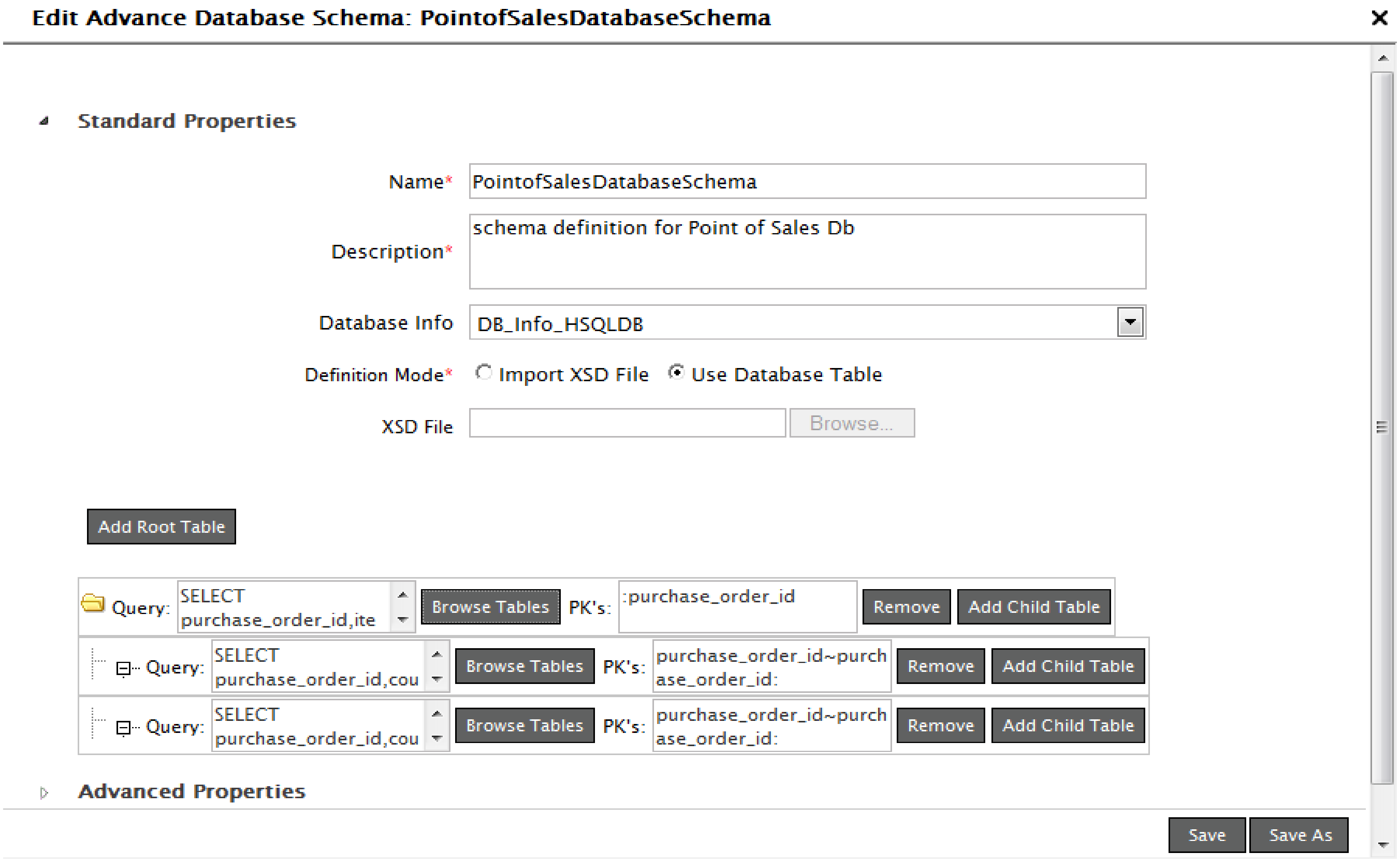
Primary Keys

AU_ID

Clear Primary Keys

© *Adeptia, Inc. All rights reserved.*

Advance Database Schema (Step 5)



Edit Advance Database Schema: PointofSalesDatabaseSchema

Standard Properties

Name*: PointofSalesDatabaseSchema

Description*: schema definition for Point of Sales Db

Database Info: DB_Info_HSQLDB

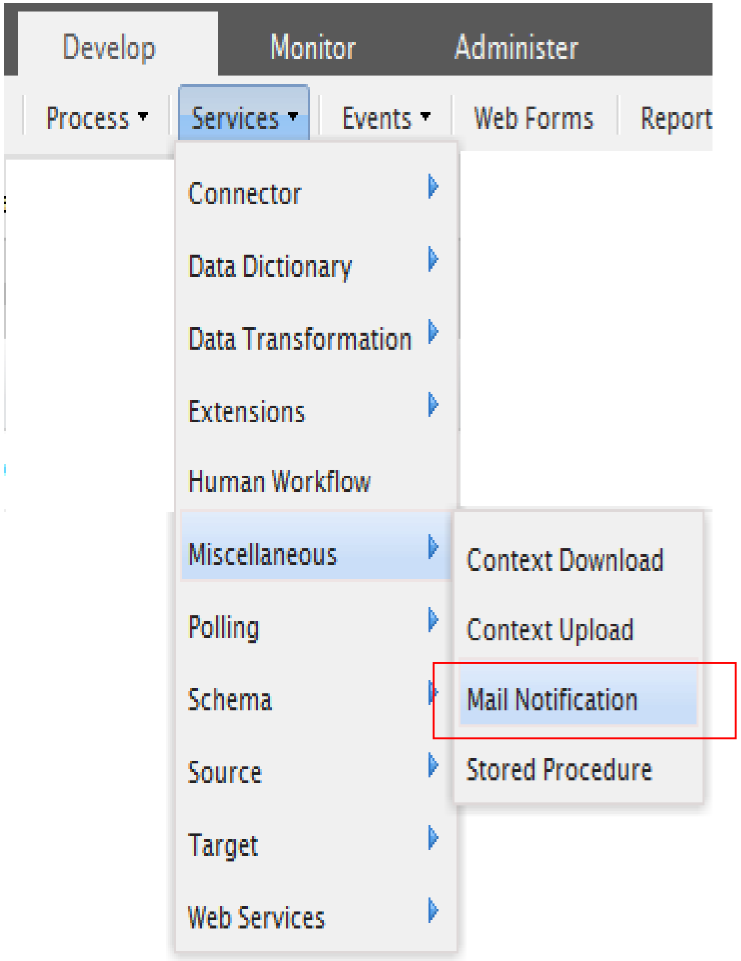
Definition Mode*: ○ Import XSD File ● Use Database Table

XSD File: Browse...

Add Root Table

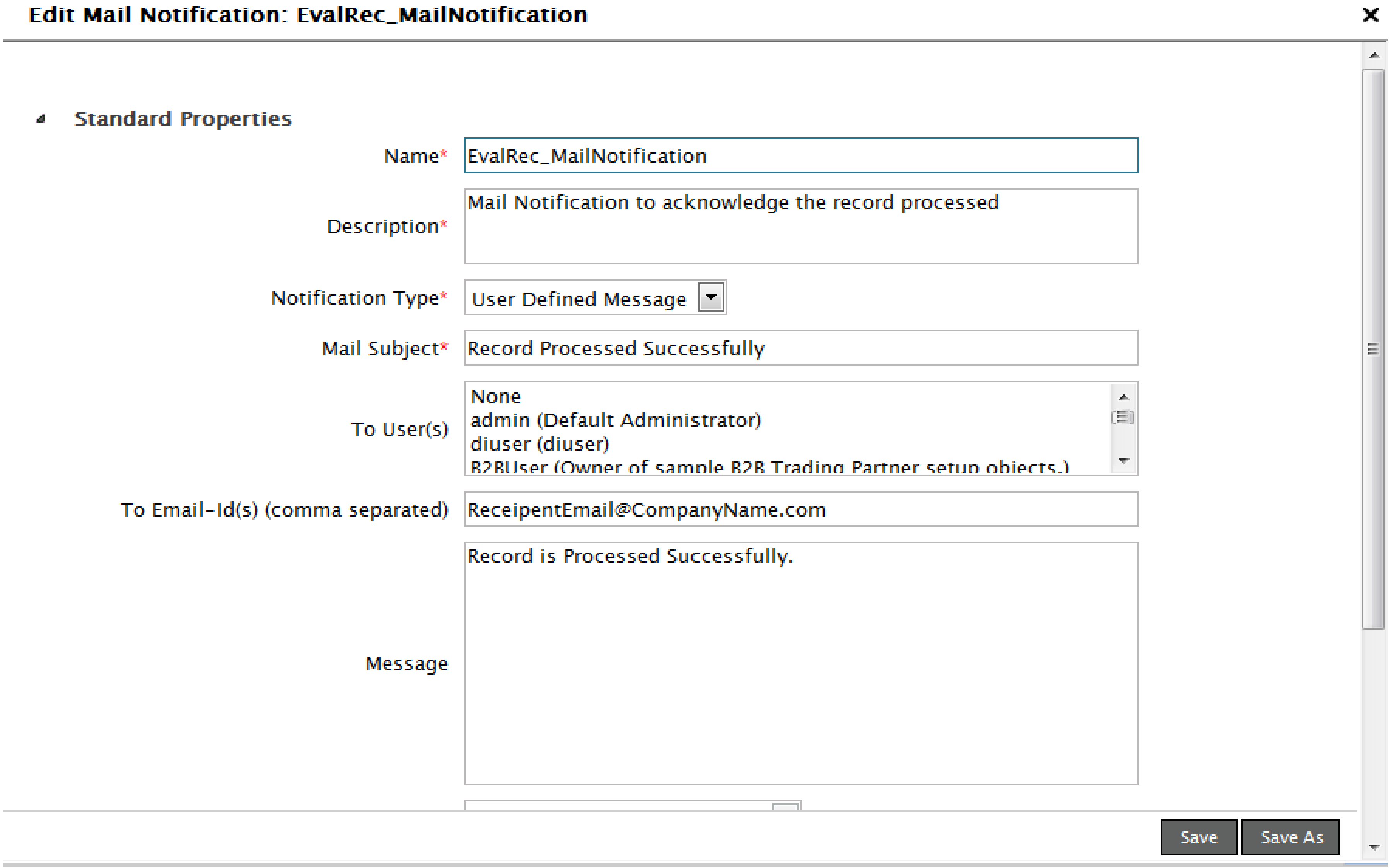
Query: SELECT purchase_order_id,ite	Browse Tables	PK's: :purchase_order_id	Remove	Add Child Table
Query: SELECT purchase_order_id,cou	Browse Tables	PK's: purchase_order_id~purchase_order_id:	Remove	Add Child Table
Query: SELECT purchase_order_id,cou	Browse Tables	PK's: purchase_order_id~purchase_order_id:	Remove	Add Child Table

Advanced Properties



Mail Notification

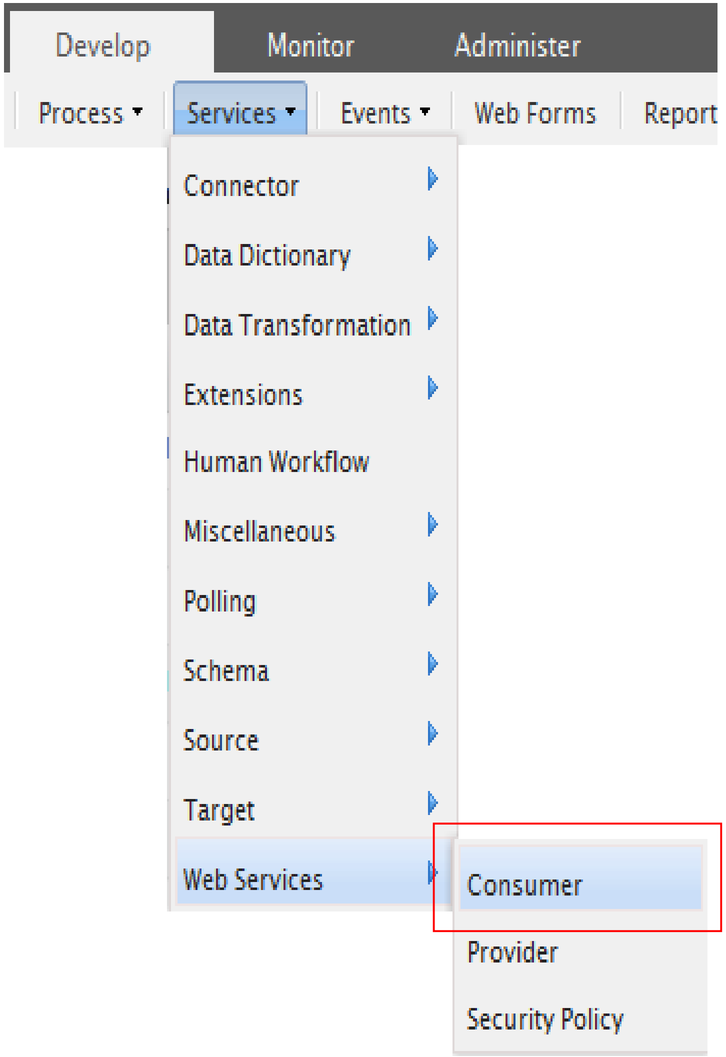
Mail Notification



Edit Mail Notification: EvalRec_MailNotification

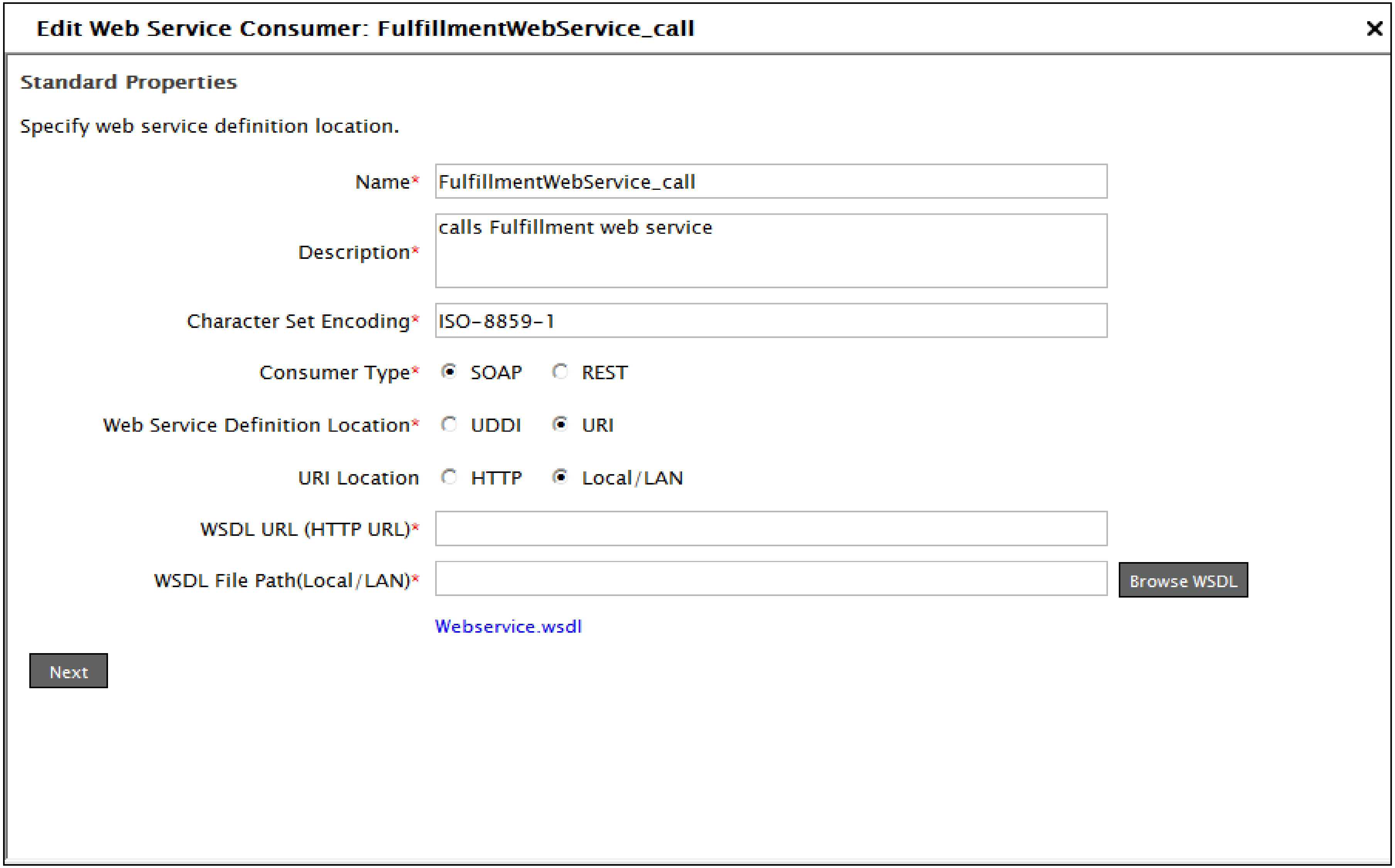
Standard Properties

Name*	EvalRec_MailNotification
Description*	Mail Notification to acknowledge the record processed
Notification Type*	User Defined Message
Mail Subject*	Record Processed Successfully
To User(s)	None admin (Default Administrator) diuser (diuser) B2BUser (Owner of sample B2B Trading Partner setup objects.)
To Email-Id(s) (comma separated)	ReceipentEmail@CompanyName.com
Message	Record is Processed Successfully.



Web Services

Web Service Consumer (Step 1)



Edit Web Service Consumer: FulfillmentWebService_call

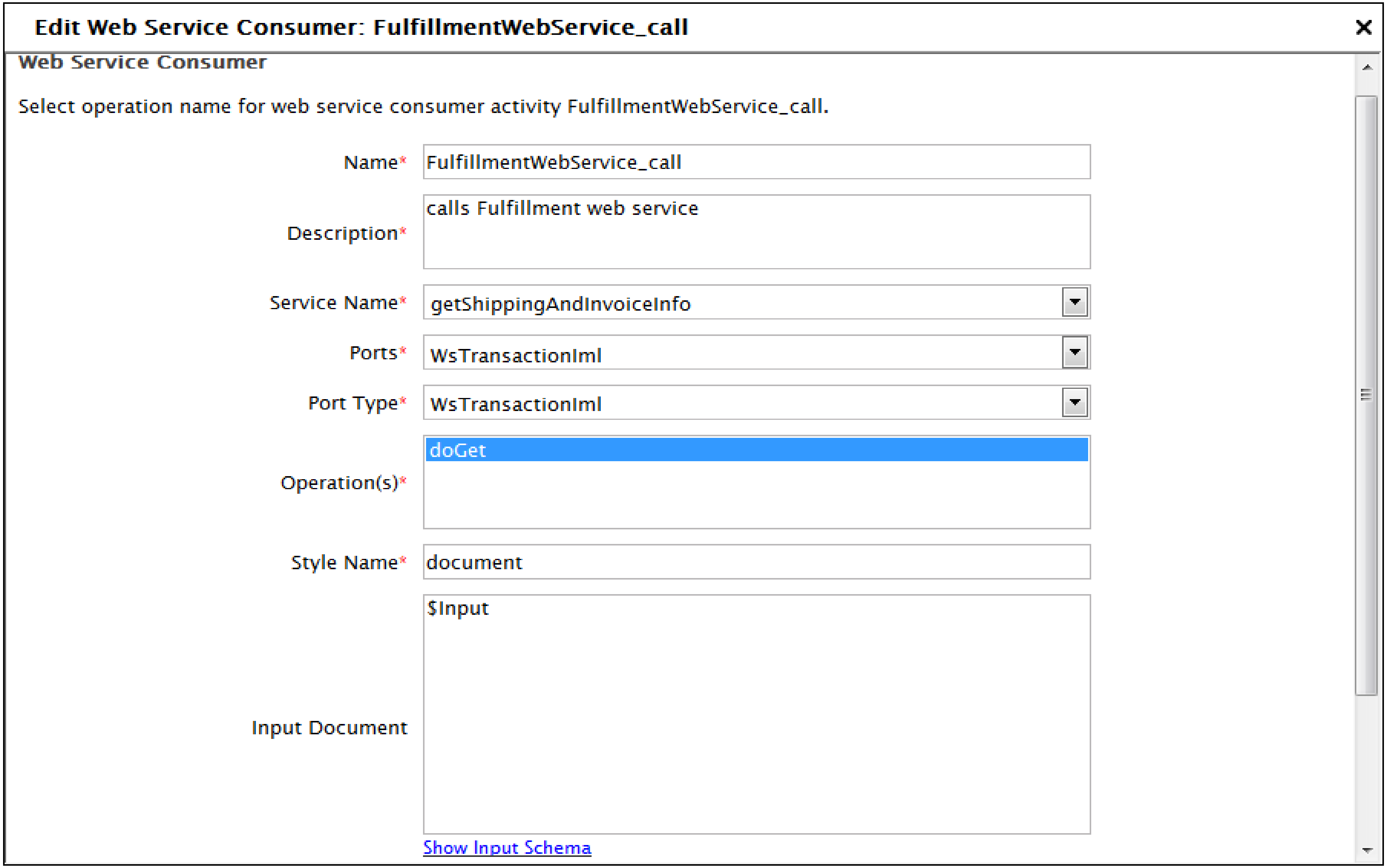
Standard Properties

Specify web service definition location.

Name*	FulfillmentWebService_call
Description*	calls Fulfillment web service
Character Set Encoding*	ISO-8859-1
Consumer Type*	● SOAP ○ REST
Web Service Definition Location*	○ UDDI ● URI
URI Location	○ HTTP ● Local/LAN
WSDL URL (HTTP URL)*	
WSDL File Path(Local/LAN)*	

Webservice.wsdl

Web Service Consumer (Step 2)



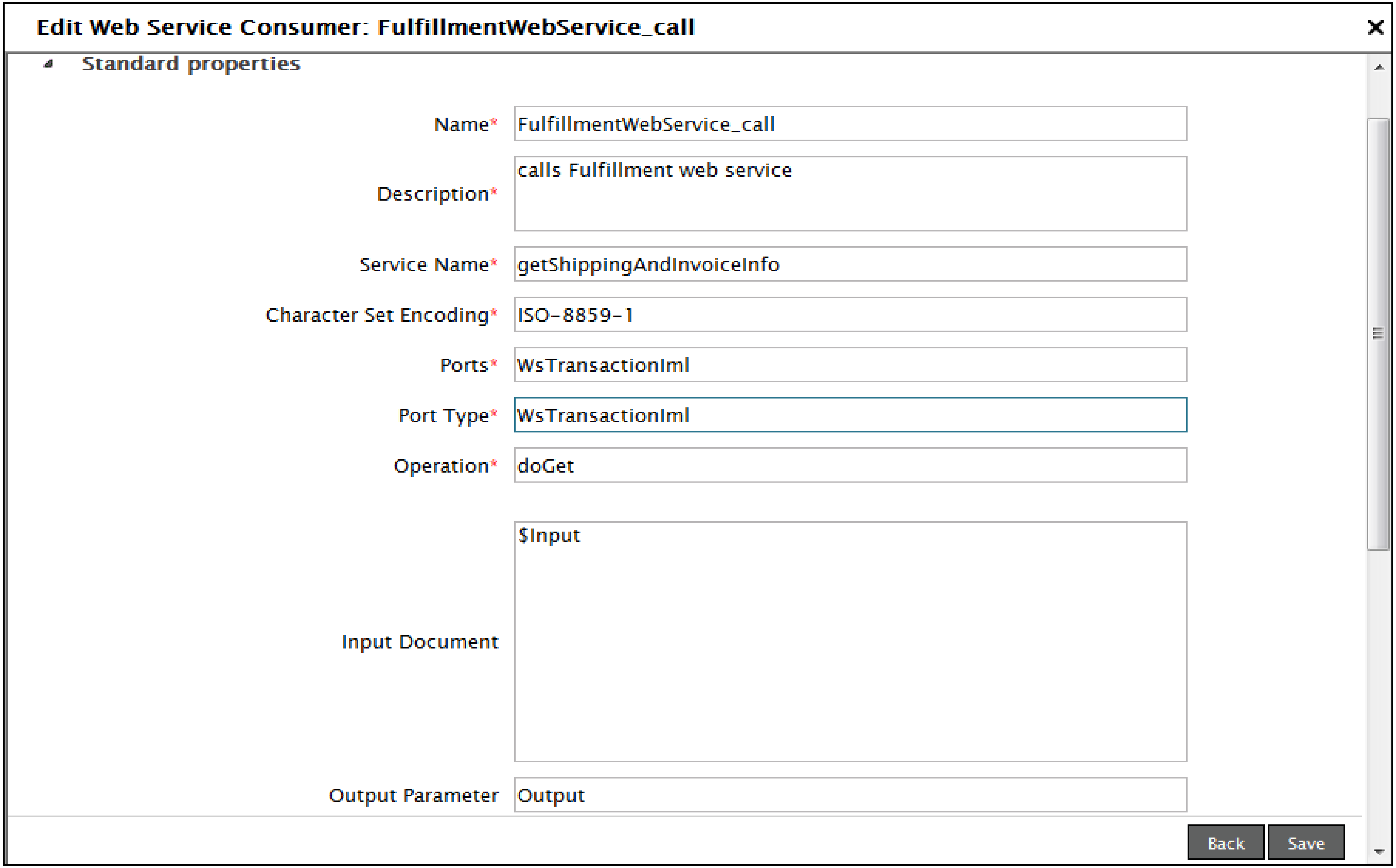
Edit Web Service Consumer: FulfillmentWebService_call

Web Service Consumer

Select operation name for web service consumer activity FulfillmentWebService_call.

Name*	FulfillmentWebService_call
Description*	calls Fulfillment web service
Service Name*	getShippingAndInvoiceInfo
Ports*	WsTransactionIml
Port Type*	WsTransactionIml
Operation(s)*	doGet
Style Name*	document
Input Document	$Input

Web Service Consumer (Step 3)

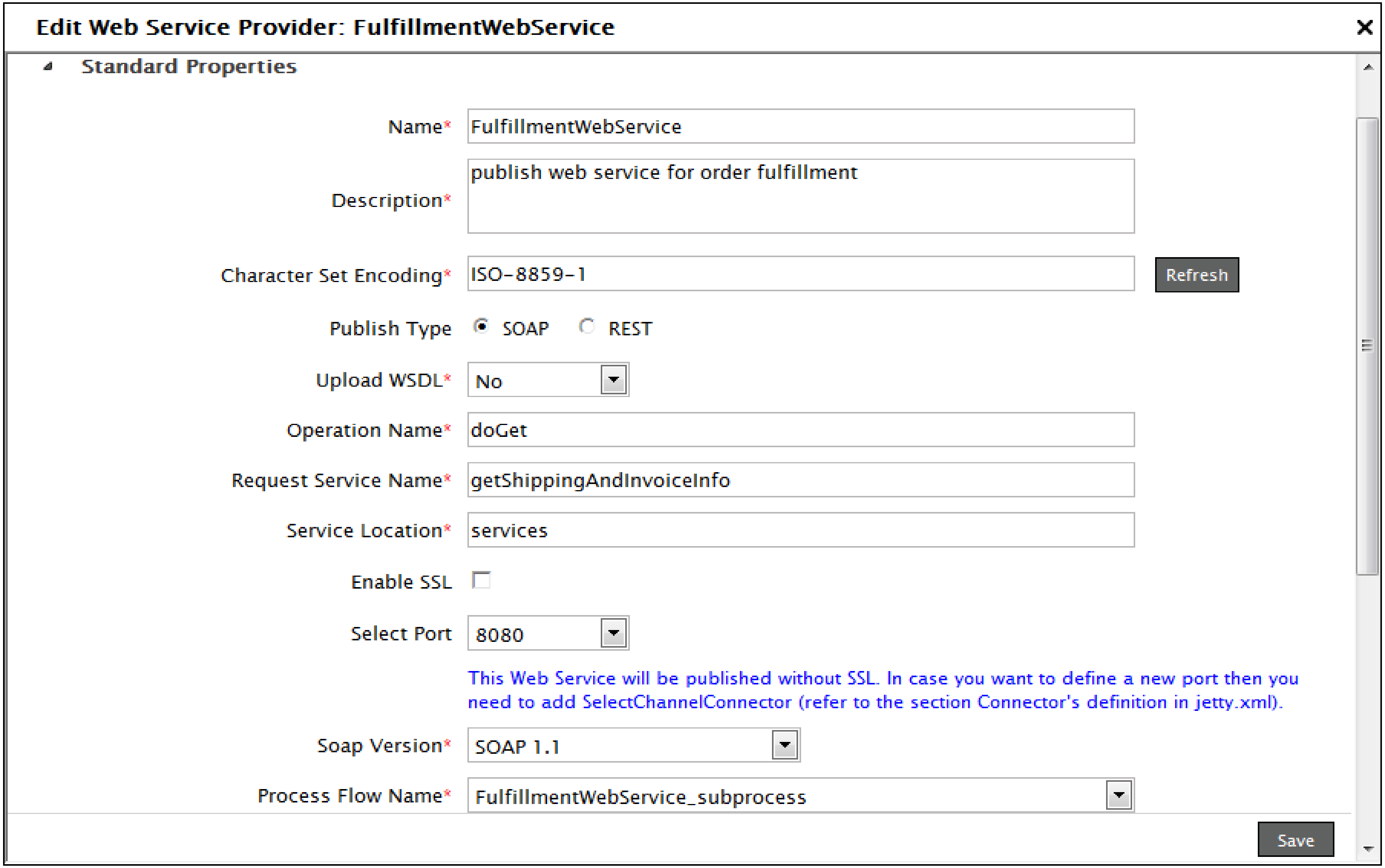


Edit Web Service Consumer: FulfillmentWebService_call

Standard properties

Name*	FulfillmentWebService_call
Description*	calls Fulfillment web service
Service Name*	getShippingAndInvoiceInfo
Character Set Encoding*	ISO-8859-1
Ports*	WsTransactionIml
Port Type*	WsTransactionIml
Operation*	doGet
Input Document	$Input
Output Parameter	Output

Web Service Provider

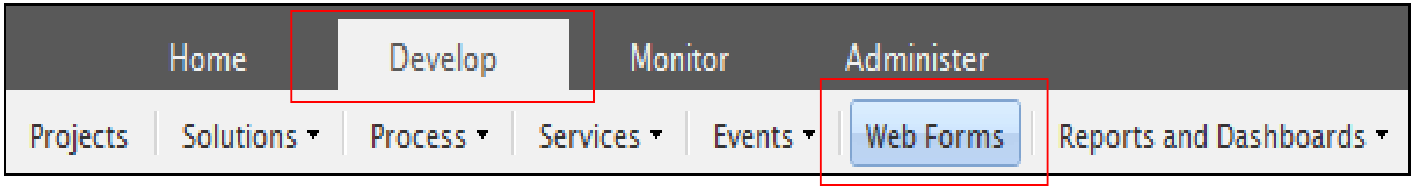


Edit Web Service Provider: FulfillmentWebService

Standard Properties

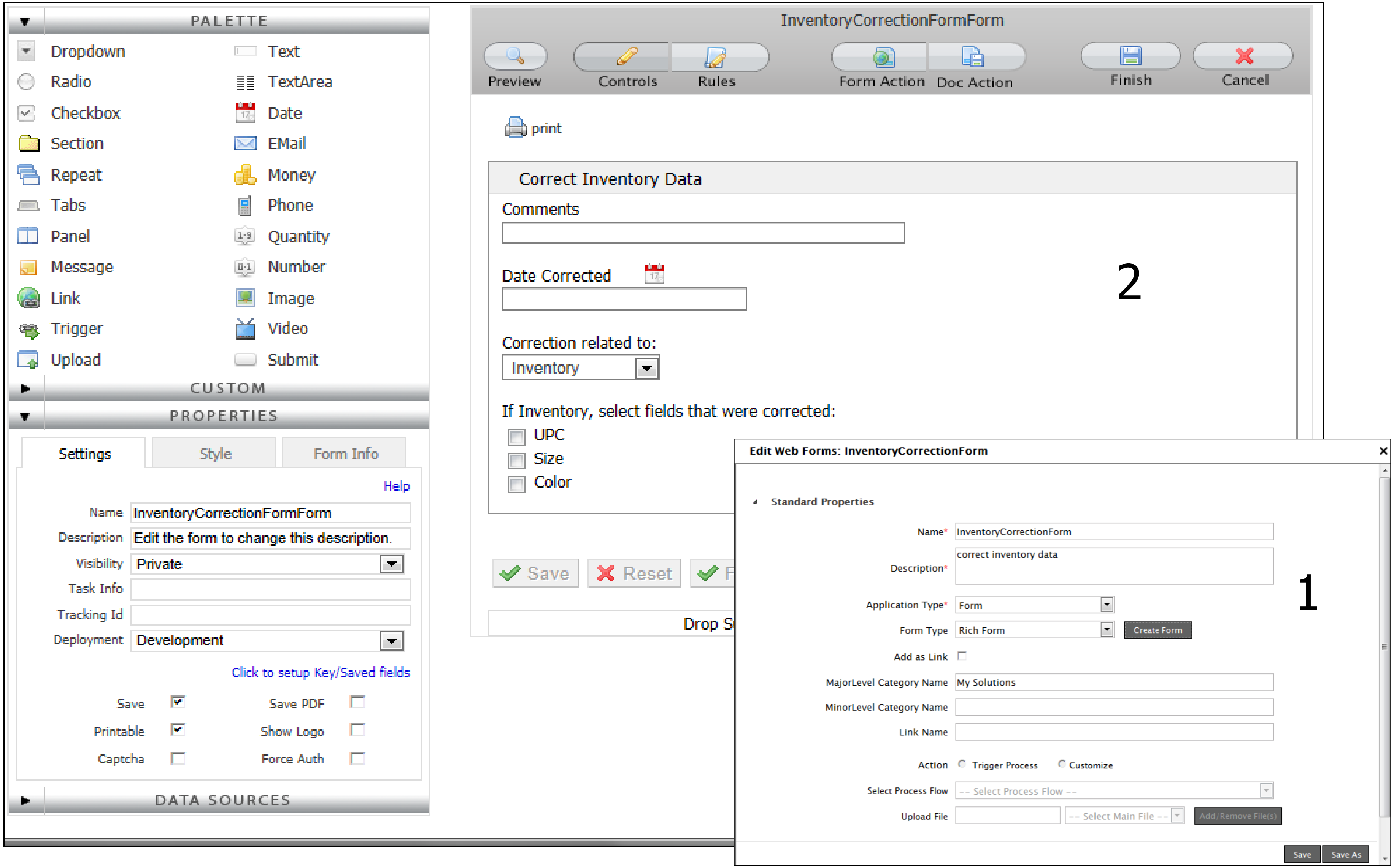
Name*	FulfillmentWebService
Description*	publish web service for order fulfillment
Character Set Encoding*	ISO-8859-1
Publish Type	● SOAP ○ REST
Upload WSDL*	No
Operation Name*	doGet
Request Service Name*	getShippingAndInvoiceInfo
Service Location*	services
Enable SSL	□
Select Port	8080
Soap Version*	SOAP 1.1
Process Flow Name*	FulfillmentWebService_subprocess

This Web Service will be published without SSL. In case you want to define a new port then you need to add SelectChannelConnector (refer to the section Connector's definition in jetty.xml).


Home
Develop
Monitor
Administer
Projects
Solutions ▾
Process ▾
Services ▾
Events ▾
Web Forms
Reports and Dashboards ▾

Web Forms

Exercise: Inventory Items Process – Part 2 (create Web Form)



Name	InventoryCorrectionFormForm
Description	Edit the form to change this description.
Visibility	Private
Task Info	
Tracking Id	
Deployment	Development

Save ☑ Save PDF □
Printable ☑ Show Logo □
Captcha □ Force Auth □

InventoryCorrectionFormForm

Correct Inventory Data

Comments

Date Corrected

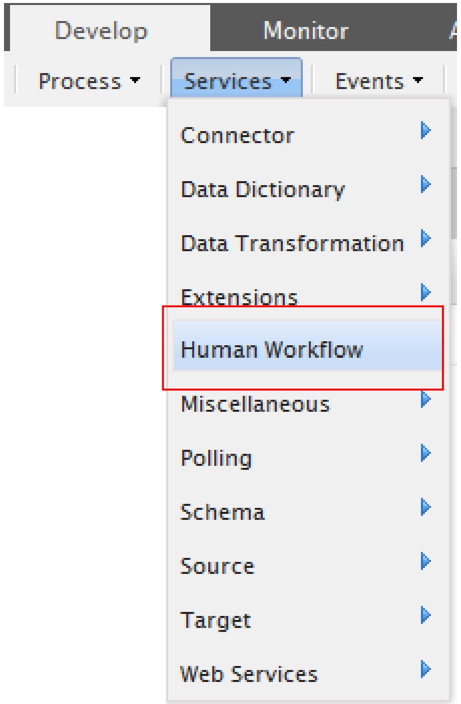
Correction related to:
Inventory

If Inventory, select fields that were corrected:
□ UPC
□ Size
□ Color

Edit Web Forms: InventoryCorrectionForm

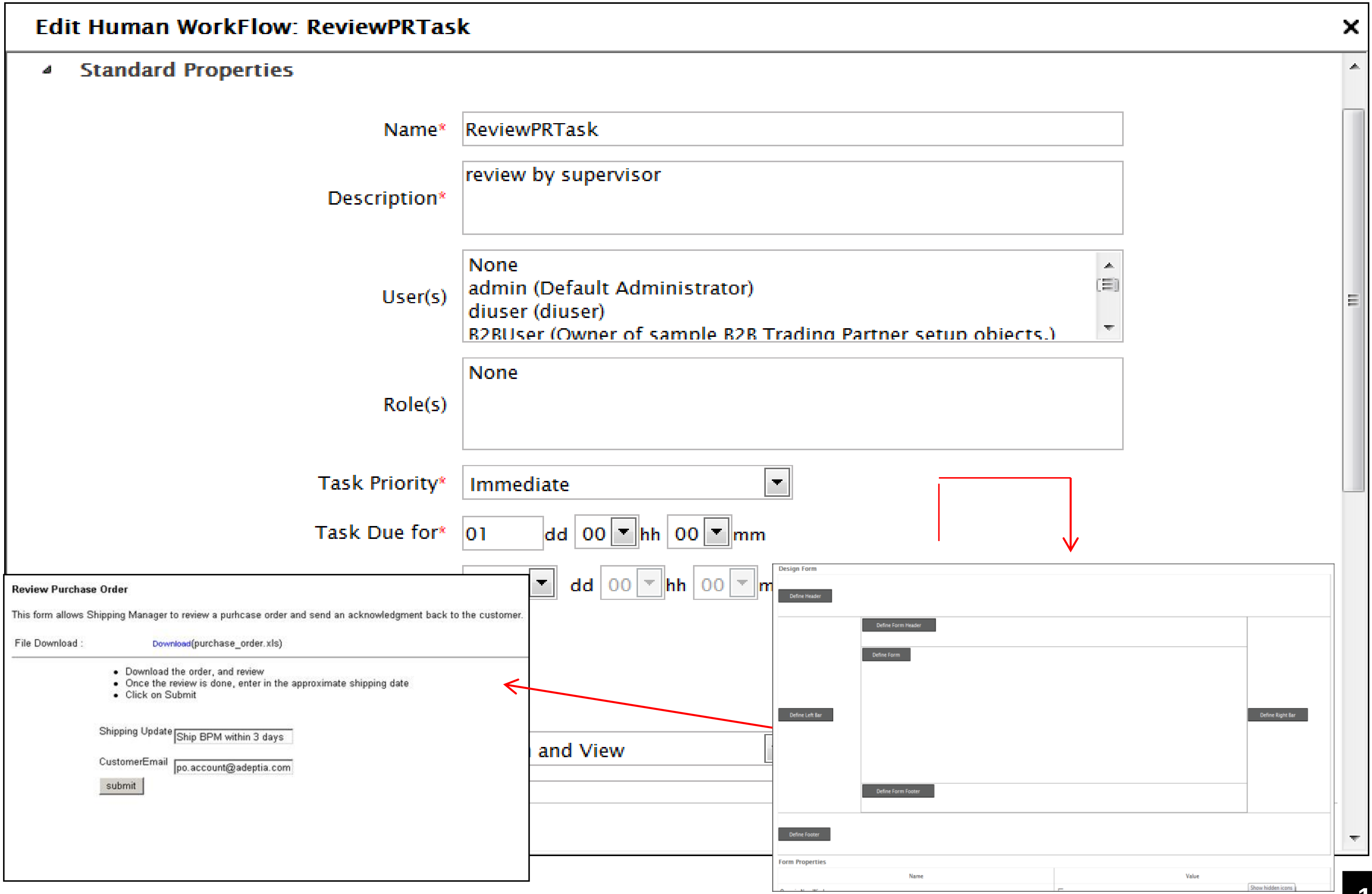
Standard Properties

Name*	InventoryCorrectionForm
Description*	correct inventory data
Application Type*	Form
Form Type	Rich Form
Add as Link	□
MajorLevel Category Name	My Solutions
MinorLevel Category Name	
Link Name	
Action	○ Trigger Process ○ Customize
Select Process Flow	-- Select Process Flow --
Upload File	-- Select Main File --



Workflow Service

Workflow Service



Edit Human WorkFlow: ReviewPRTask

Standard Properties

Name*	ReviewPRTask
Description*	review by supervisor
User(s)	None admin (Default Administrator) diuser (diuser) B2BUser (Owner of sample B2B Trading Partner setup objects.)
Role(s)	None
Task Priority*	Immediate
Task Due for*	01 dd 00 hh 00 mm

dd 00 hh 00 m

and View

Review Purchase Order

This form allows Shipping Manager to review a purhcase order and send an acknowledgment back to the customer.

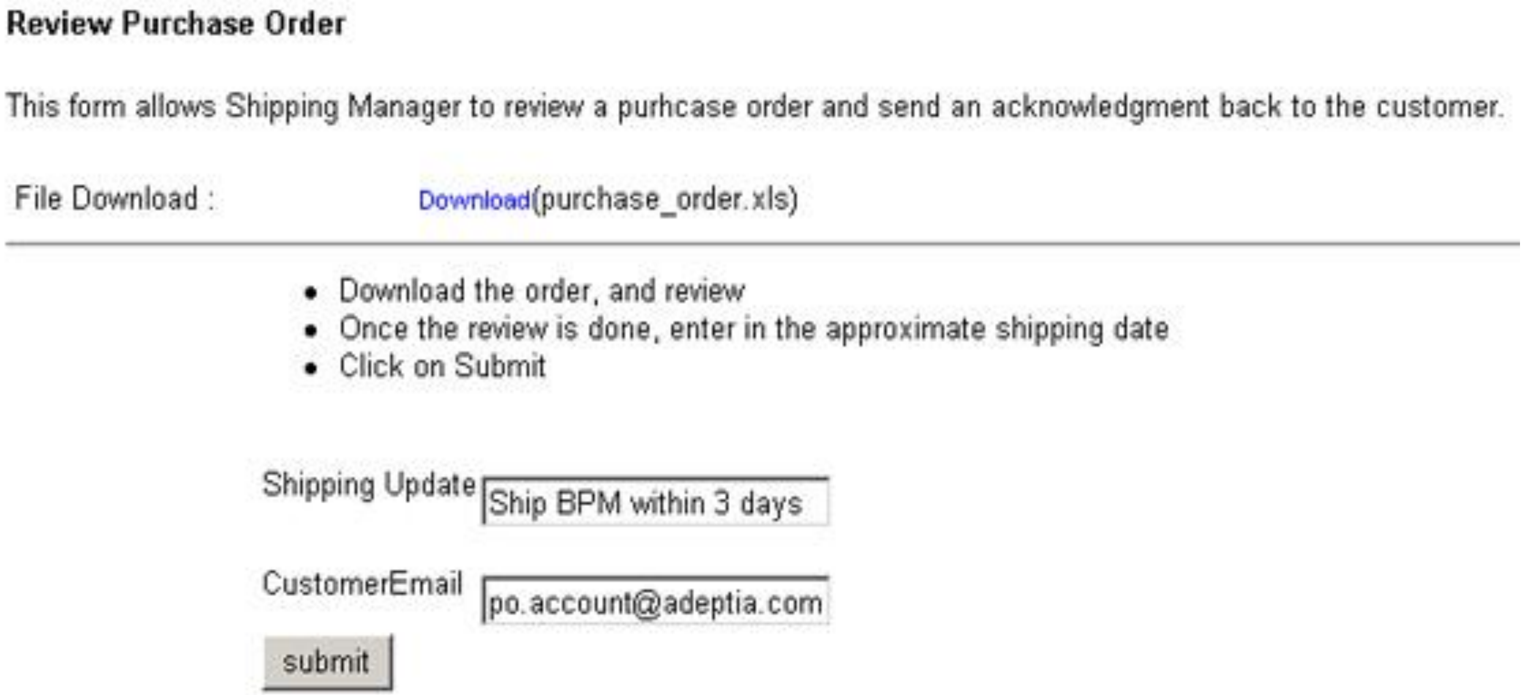
File Download : Download(purchase_order.xls)

- Download the order, and review
- Once the review is done, enter in the approximate shipping date
- Click on Submit

Shipping Update	Ship BPM within 3 days
CustomerEmail	po.account@adeptia.com

submit

Exercise: Create Workflow Activity



Review Purchase Order

This form allows Shipping Manager to review a purhcase order and send an acknowledgment back to the customer.

File Download : Download(purchase_order.xls)

- Download the order, and review
- Once the review is done, enter in the approximate shipping date
- Click on Submit

Shipping Update: Ship BPM within 3 days

CustomerEmail: po.account@adeptia.com

Training schedule

Day 1

Morning session
9am to 12pm, 10 min break at 10:30am

- Introduction
  - Overview of Adeptia Suite
  - Architecture
  - Project approach
- Process Modeler & Simulation (BPM)
  - Modeler
  - Simulator
- Process Designer
- BPM Exercise – Purchase Requisition Workflow (BPM)
  - Web Forms
  - Workflow

Afternoon session
1pm to 4:30pm, 10 min break @ 2:20pm and 3:20pm

- Integration/ETL Exercises:
  - Employee Benefits
  - Purchase Order
  - Inventory Items
  - Order Fulfillment
- Services: Description and Exercises
  - Sources
  - Targets
  - Data Dictionary
  - Schema
  - Mapping
  - Web Services
  - Workflow
- Events & Triggers

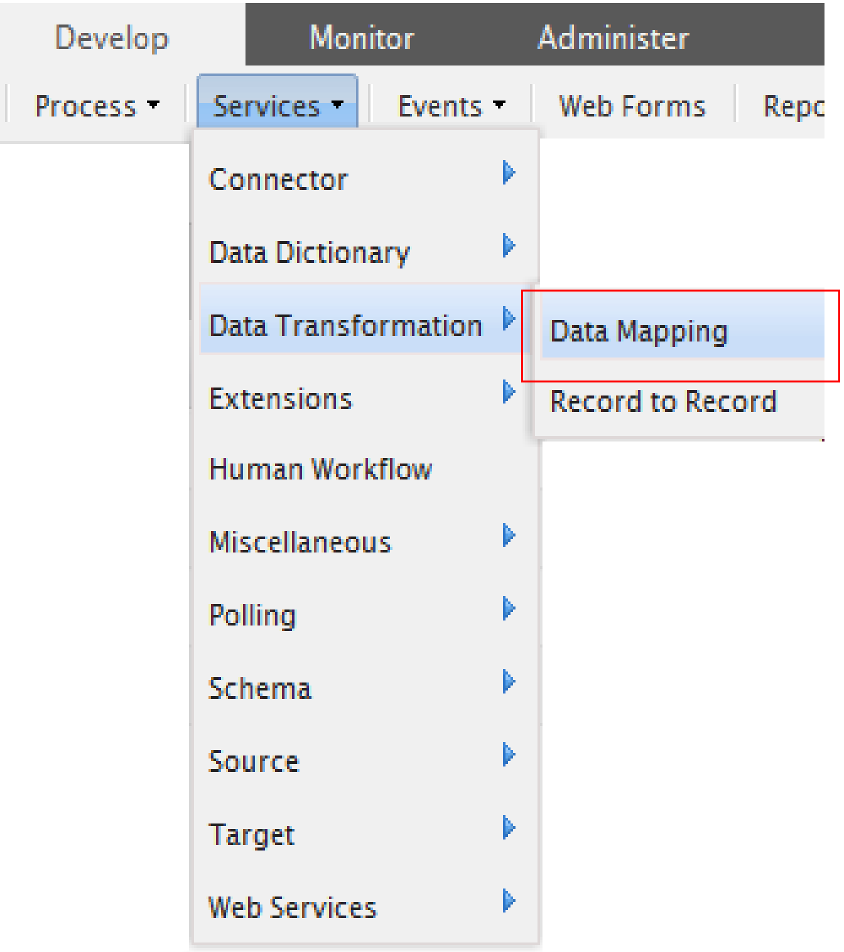
Day 2

Morning session
9am to 12pm, 10 min break at 10:30am

- Advanced Mapping & Process Designer functions
  - Process Designer
    - Decision nodes
    - Context Variables
    - Sub Processes
  - Mapping
    - Math, String and Conditional functions
    - Database lookups
    - Context variables
    - Multiple Source and Target mapping
- Execute flow
  - Trigger
  - Logs
  - Reports

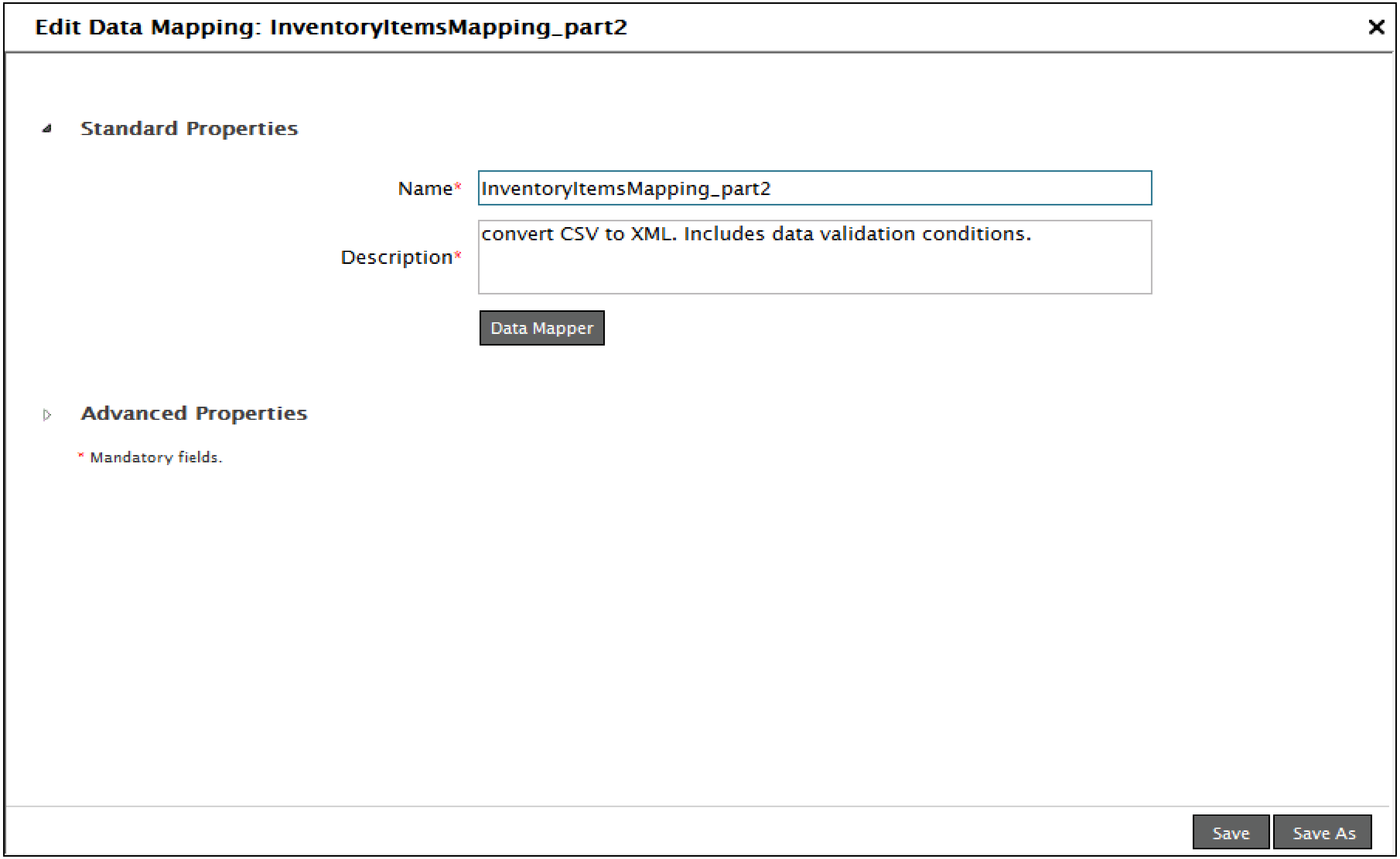
Afternoon session
1pm to 4pm, 10 min break @ 2:30pm

- Managing Solutions
  - My Tasks and Task History
  - Solution and Performance Monitoring
  - Logs
- Administration
  - User management
  - Configuration and Properties
  - System monitoring
  - Maintenance activities



Data Mapping Service

Mapping (Selecting Source/Target Schemas)



Edit Data Mapping: InventoryItemsMapping_part2

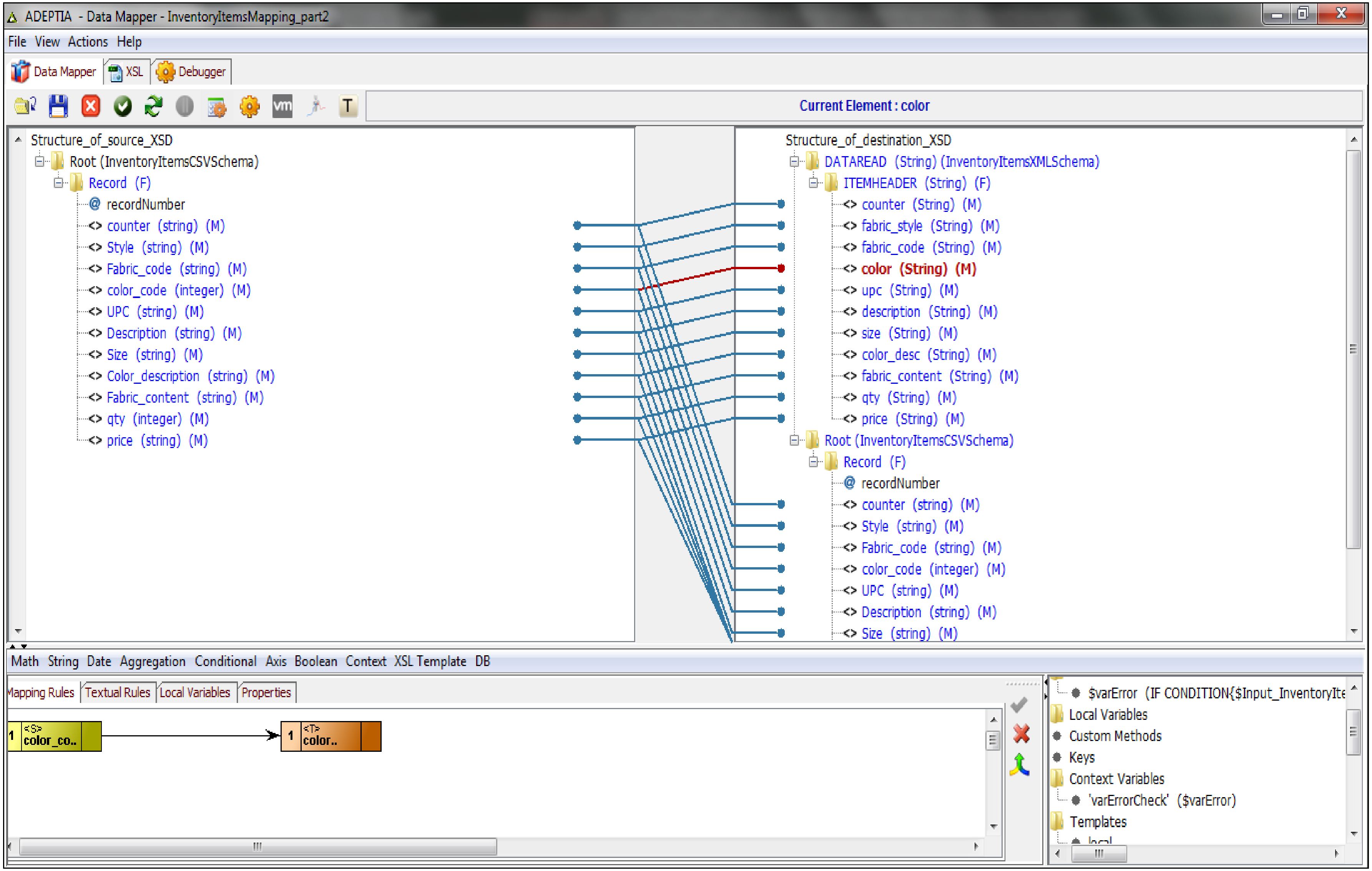
Standard Properties

Name*: InventoryItemsMapping_part2

Description*: convert CSV to XML. Includes data validation conditions.

* Mandatory fields.

Mapping (Auto Mapper)



ADEPTIA - Data Mapper - InventoryItemsMapping_part2

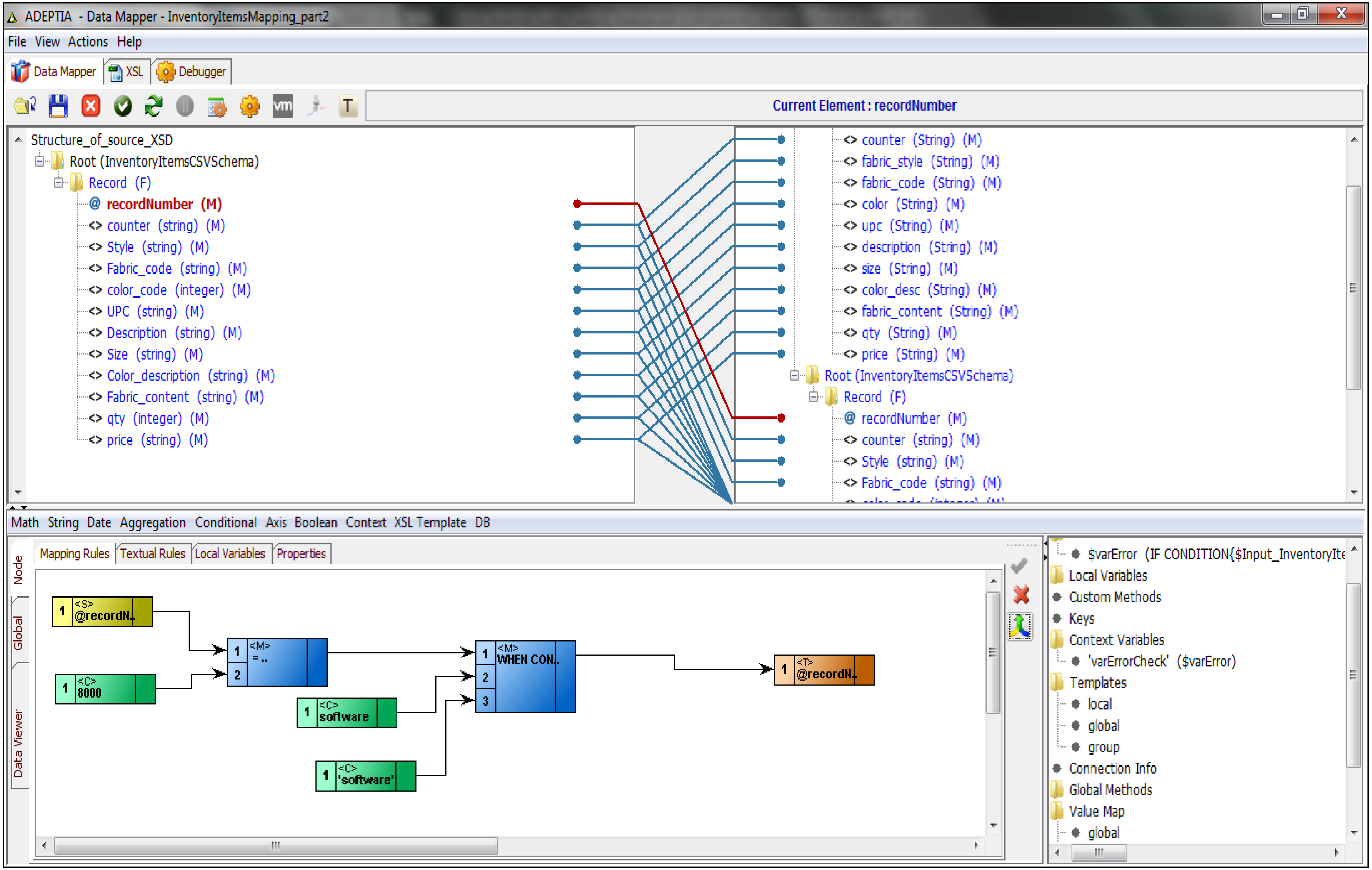
Current Element : color

Structure_of_source_XSD
Root (InventoryItemsCSVSchema)
Record (F)
@ recordNumber
<> counter (string) (M)
<> Style (string) (M)
<> Fabric_code (string) (M)
<> color_code (integer) (M)
<> UPC (string) (M)
<> Description (string) (M)
<> Size (string) (M)
<> Color_description (string) (M)
<> Fabric_content (string) (M)
<> qty (integer) (M)
<> price (string) (M)

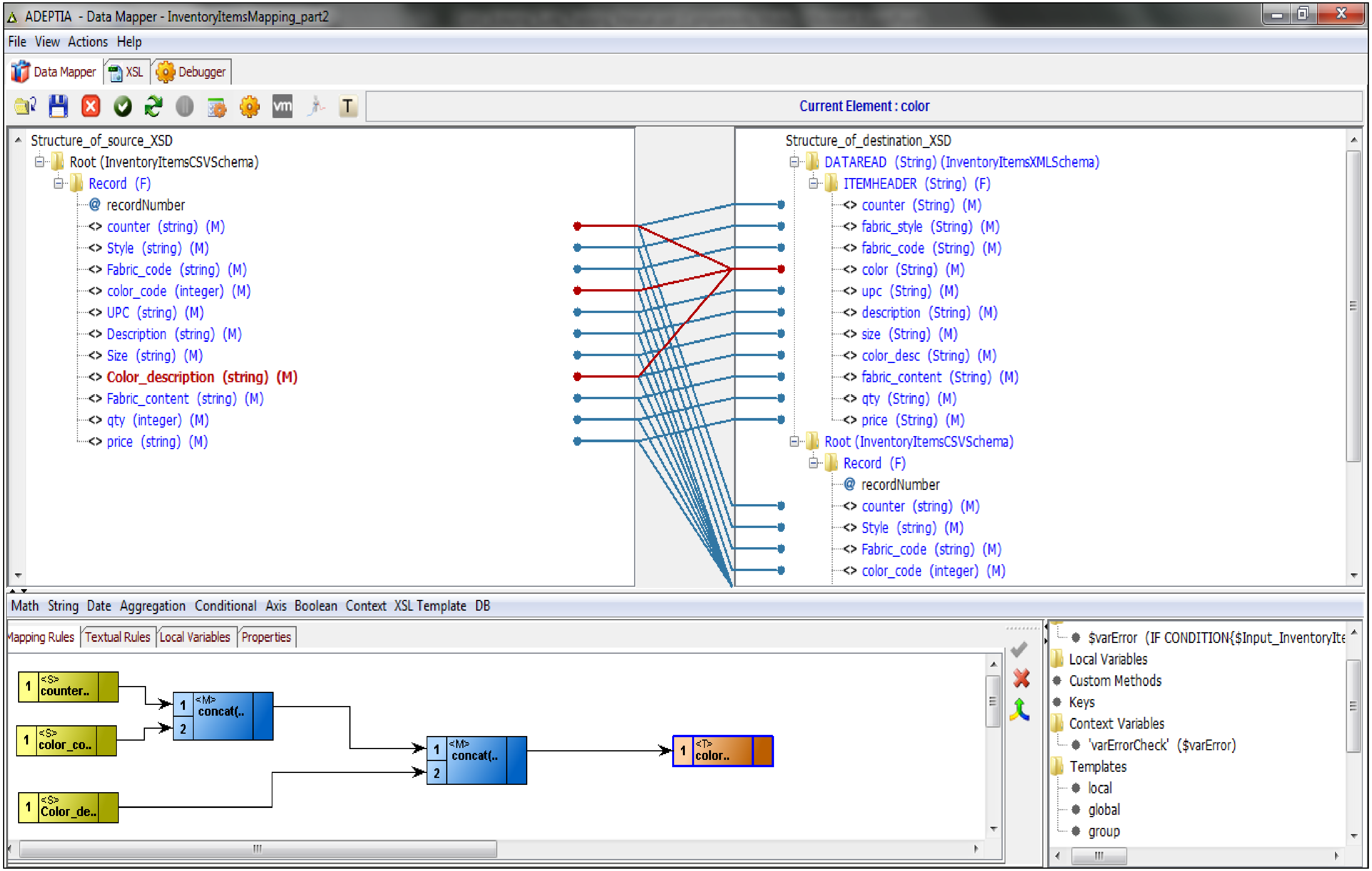
Structure_of_destination_XSD
DATAREAD (String) (InventoryItemsXMLSchema)
ITEMHEADER (String) (F)
<> counter (String) (M)
<> fabric_style (String) (M)
<> fabric_code (String) (M)
<> color (String) (M)
<> upc (String) (M)
<> description (String) (M)
<> size (String) (M)
<> color_desc (String) (M)
<> fabric_content (String) (M)
<> qty (String) (M)
<> price (String) (M)
Root (InventoryItemsCSVSchema)
Record (F)
@ recordNumber
<> counter (string) (M)
<> Style (string) (M)
<> Fabric_code (string) (M)
<> color_code (integer) (M)
<> UPC (string) (M)
<> Description (string) (M)
<> Size (string) (M)

$varError (IF CONDITION{$Input_InventoryIte
Local Variables
Custom Methods
Keys
Context Variables
'varErrorCheck' ($varError)
Templates
local

Mapping (Using When Condition)



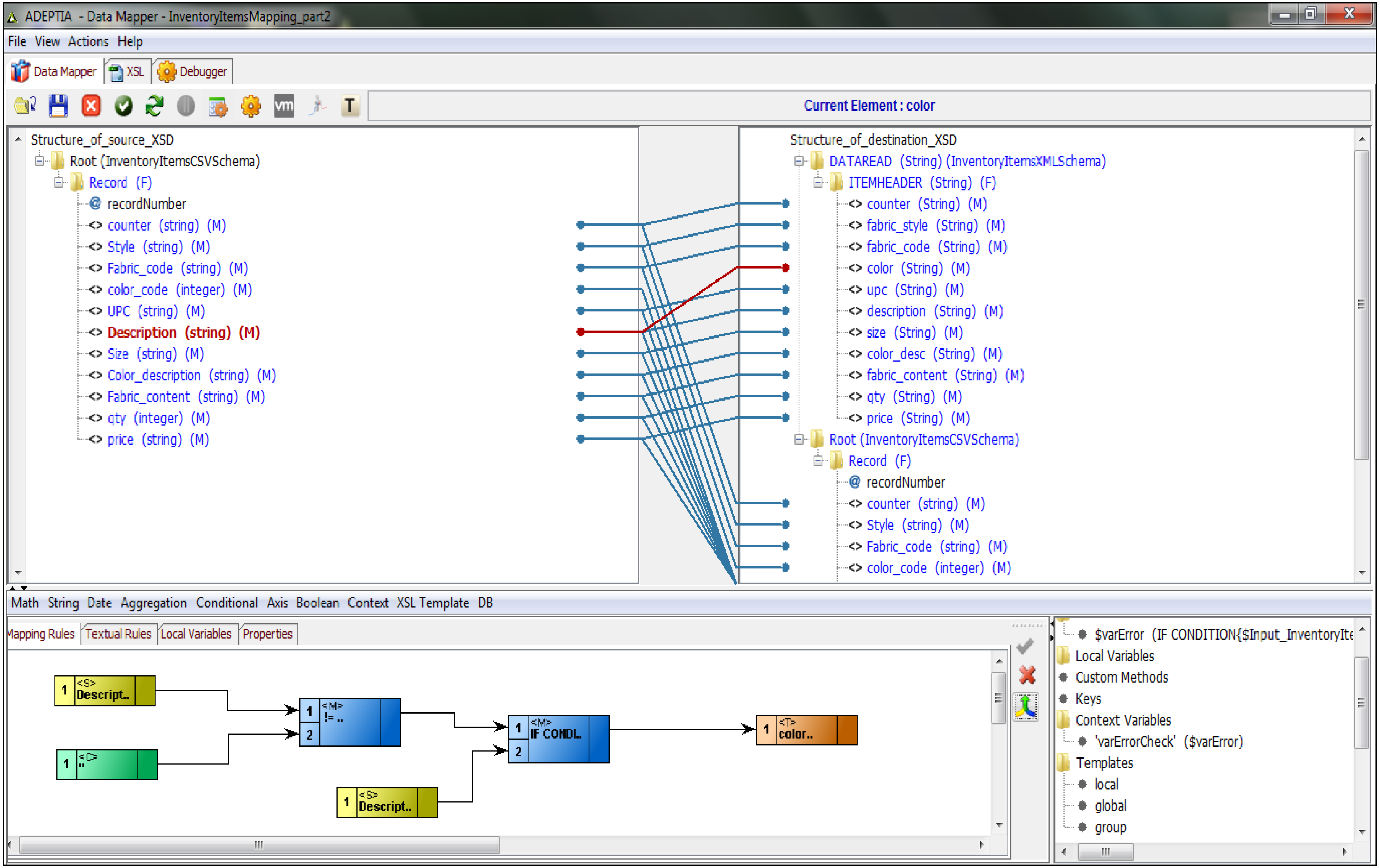
Mapping (Using String-Concat operator)



ADEPTIA - Data Mapper - InventoryItemsMapping_part2
File View Actions Help
Data Mapper
XSL
Debugger
vm
T
Structure_of_source_XSD
Root (InventoryItemsCSVSchema)
Record (F)
@ recordNumber
<> counter (string) (M)
<> Style (string) (M)
<> Fabric_code (string) (M)
<> color_code (integer) (M)
<> UPC (string) (M)
<> Description (string) (M)
<> Size (string) (M)
<> Color_description (string) (M)
<> Fabric_content (string) (M)
<> qty (integer) (M)
<> price (string) (M)
Structure_of_destination_XSD
DATAREAD (String) (InventoryItemsXMLSchema)
ITEMHEADER (String) (F)
<> counter (String) (M)
<> fabric_style (String) (M)
<> fabric_code (String) (M)
<> color (String) (M)
<> upc (String) (M)
<> description (String) (M)
<> size (String) (M)
<> color_desc (String) (M)
<> fabric_content (String) (M)
<> qty (String) (M)
<> price (String) (M)
Root (InventoryItemsCSVSchema)
Record (F)
@ recordNumber
<> counter (string) (M)
<> Style (string) (M)
<> Fabric_code (string) (M)
<> color_code (integer) (M)
Math String Date Aggregation Conditional Axis Boolean Context XSL Template DB
Mapping Rules
Textual Rules
Local Variables
Properties
1 <S> counter..
1 <S> color_co..
1 <S> Color_de..
1 2 <M> concat(..
1 2 <M> concat(..
1 <T> color..
$varError (IF CONDITION{$Input_InventoryIte
Local Variables
Custom Methods
Keys
Context Variables
'varErrorCheck' ($varError)
Templates
local
global
group

Current Element : color

Mapping (Using IFF Condition)



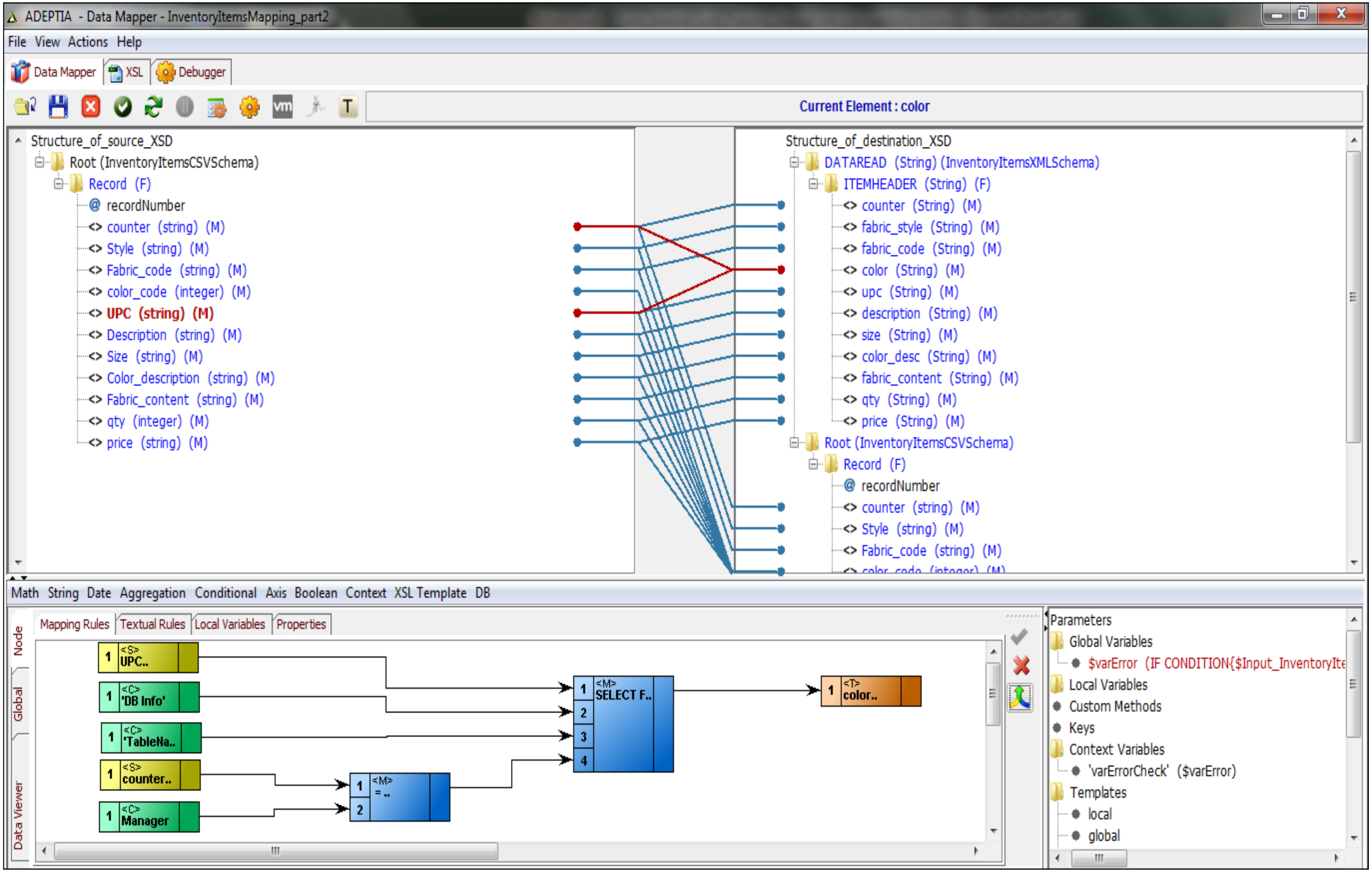
Current Element : color

Structure_of_source_XSD
Root (InventoryItemsCSVSchema)
Record (F)
@ recordNumber
<> counter (string) (M)
<> Style (string) (M)
<> Fabric_code (string) (M)
<> color_code (integer) (M)
<> UPC (string) (M)
<> Description (string) (M)
<> Size (string) (M)
<> Color_description (string) (M)
<> Fabric_content (string) (M)
<> qty (integer) (M)
<> price (string) (M)

Structure_of_destination_XSD
DATAREAD (String) (InventoryItemsXMLSchema)
ITEMHEADER (String) (F)
<> counter (String) (M)
<> fabric_style (String) (M)
<> fabric_code (String) (M)
<> color (String) (M)
<> upc (String) (M)
<> description (String) (M)
<> size (String) (M)
<> color_desc (String) (M)
<> fabric_content (String) (M)
<> qty (String) (M)
<> price (String) (M)
Root (InventoryItemsCSVSchema)
Record (F)
@ recordNumber
<> counter (string) (M)
<> Style (string) (M)
<> Fabric_code (string) (M)
<> color_code (integer) (M)

$varError (IF CONDITION{$Input_InventoryIte
Local Variables
Custom Methods
Keys
Context Variables
'varErrorCheck' ($varError)
Templates
local
global
group

Mapping (Using Database Lookup)



Structure_of_source_XSD

- Root (InventoryItemsCSVSchema)
  - Record (F)
    - @ recordNumber
    - <> counter (string) (M)
    - <> Style (string) (M)
    - <> Fabric_code (string) (M)
    - <> color_code (integer) (M)
    - <> UPC (string) (M)
    - <> Description (string) (M)
    - <> Size (string) (M)
    - <> Color_description (string) (M)
    - <> Fabric_content (string) (M)
    - <> qty (integer) (M)
    - <> price (string) (M)

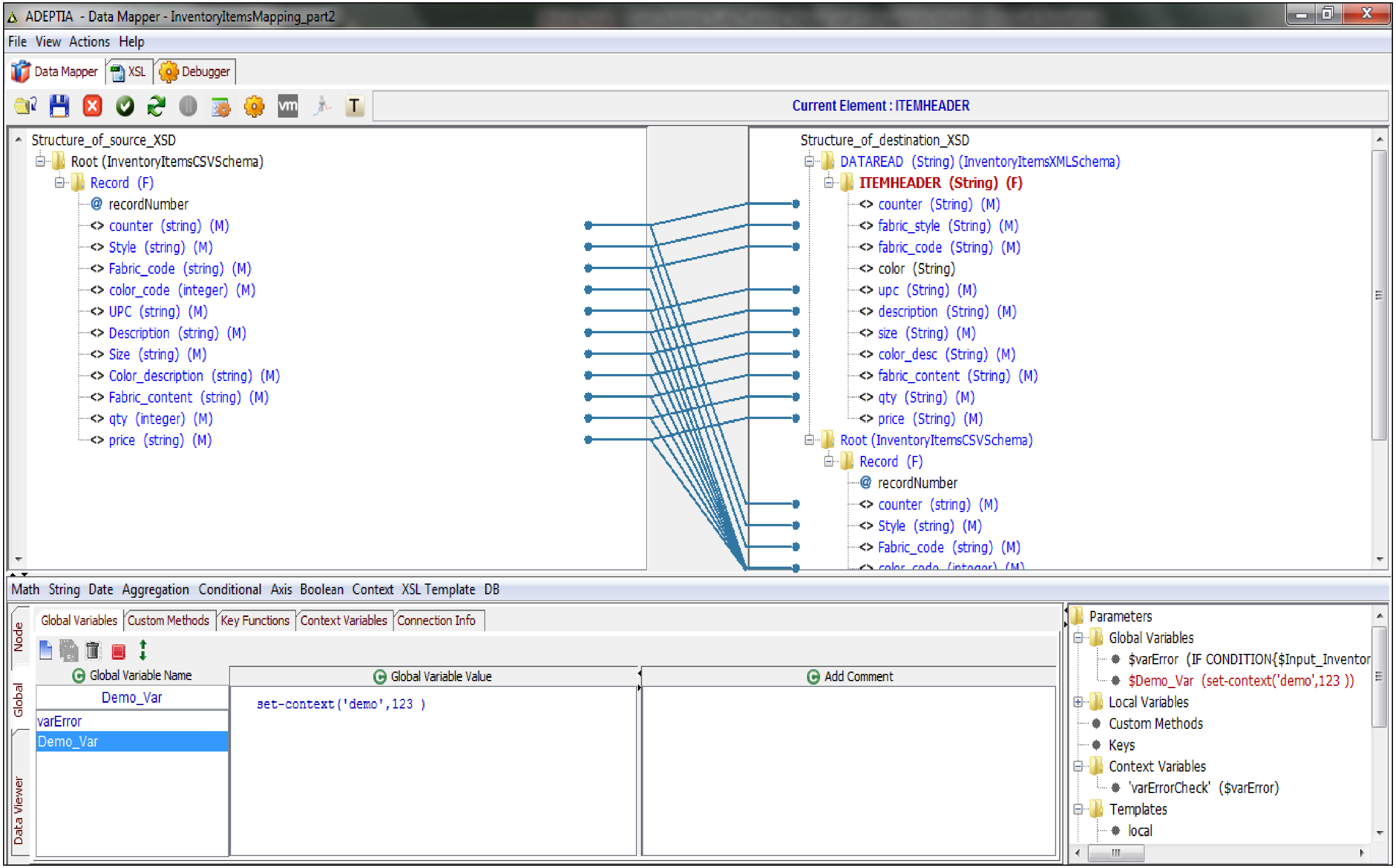
Structure_of_destination_XSD

- DATAREAD (String) (InventoryItemsXMLSchema)
  - ITEMHEADER (String) (F)
    - <> counter (String) (M)
    - <> fabric_style (String) (M)
    - <> fabric_code (String) (M)
    - <> color (String) (M)
    - <> upc (String) (M)
    - <> description (String) (M)
    - <> size (String) (M)
    - <> color_desc (String) (M)
    - <> fabric_content (String) (M)
    - <> qty (String) (M)
    - <> price (String) (M)
- Root (InventoryItemsCSVSchema)
  - Record (F)
    - @ recordNumber
    - <> counter (string) (M)
    - <> Style (string) (M)
    - <> Fabric_code (string) (M)
    - <> color_code (integer) (M)

Parameters

- Global Variables
  - $varError (IF CONDITION{$Input_InventoryIte
- Local Variables
- Custom Methods
- Keys
- Context Variables
  - 'varErrorCheck' ($varError)
- Templates
  - local
  - global

Mapping (Global Variable – Set Context)



ADEPTIA - Data Mapper - InventoryItemsMapping_part2

File View Actions Help

Data Mapper | XSL | Debugger

Current Element : ITEMHEADER

Structure_of_source_XSD
- Root (InventoryItemsCSVSchema)
  - Record (F)
    - @ recordNumber
    - <> counter (string) (M)
    - <> Style (string) (M)
    - <> Fabric_code (string) (M)
    - <> color_code (integer) (M)
    - <> UPC (string) (M)
    - <> Description (string) (M)
    - <> Size (string) (M)
    - <> Color_description (string) (M)
    - <> Fabric_content (string) (M)
    - <> qty (integer) (M)
    - <> price (string) (M)

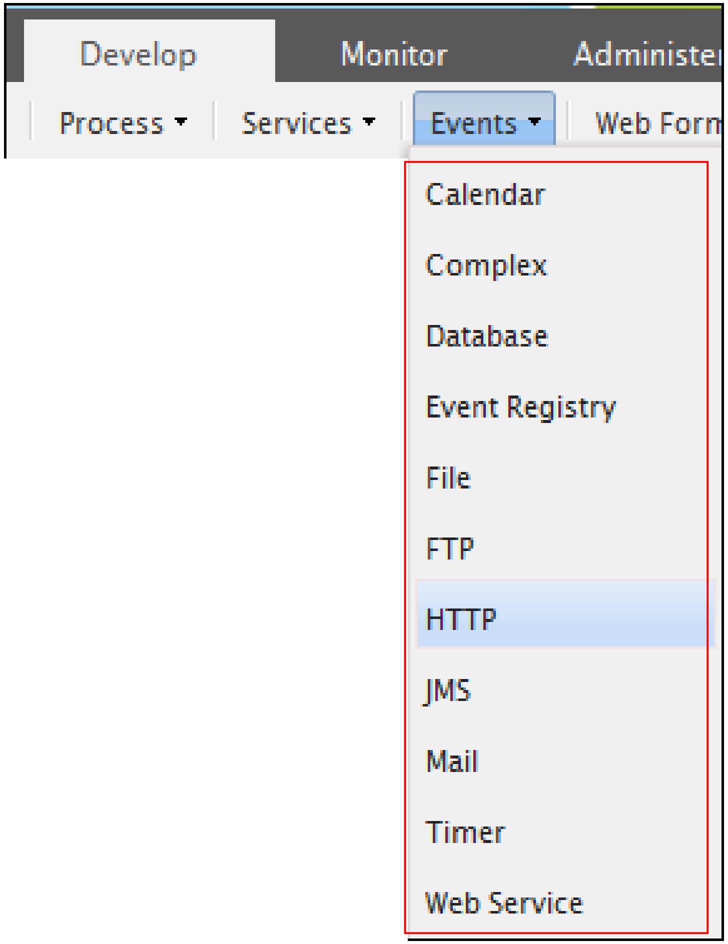
Structure_of_destination_XSD
- DATAREAD (String) (InventoryItemsXMLSchema)
  - ITEMHEADER (String) (F)
    - <> counter (String) (M)
    - <> fabric_style (String) (M)
    - <> fabric_code (String) (M)
    - <> color (String)
    - <> upc (String) (M)
    - <> description (String) (M)
    - <> size (String) (M)
    - <> color_desc (String) (M)
    - <> fabric_content (String) (M)
    - <> qty (String) (M)
    - <> price (String) (M)
- Root (InventoryItemsCSVSchema)
  - Record (F)
    - @ recordNumber
    - <> counter (string) (M)
    - <> Style (string) (M)
    - <> Fabric_code (string) (M)
    - <> color_code (integer) (M)

Math String Date Aggregation Conditional Axis Boolean Context XSL Template DB

Global Variables | Custom Methods | Key Functions | Context Variables | Connection Info

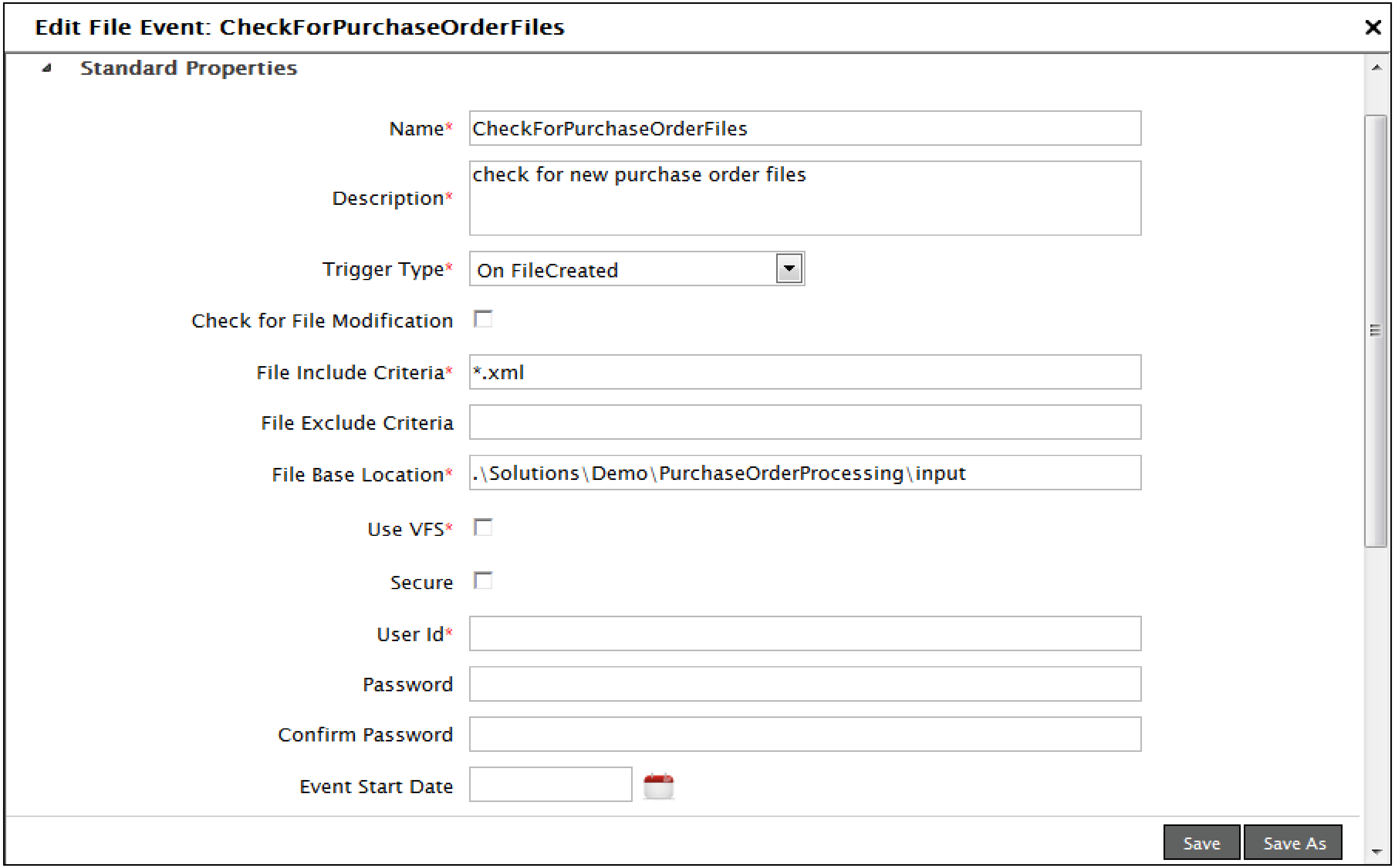
Global Variable Name	Global Variable Value	Add Comment
Demo_Var	`set-context('demo',123 )`	
varError		
Demo_Var		

- Parameters
- Global Variables
  - $varError (IF CONDITION{$Input_Inventor
  - $Demo_Var (set-context('demo',123))
- Local Variables
- Custom Methods
- Keys
- Context Variables
  - 'varErrorCheck' ($varError)
- Templates
  - local



Event Triggers

File Event

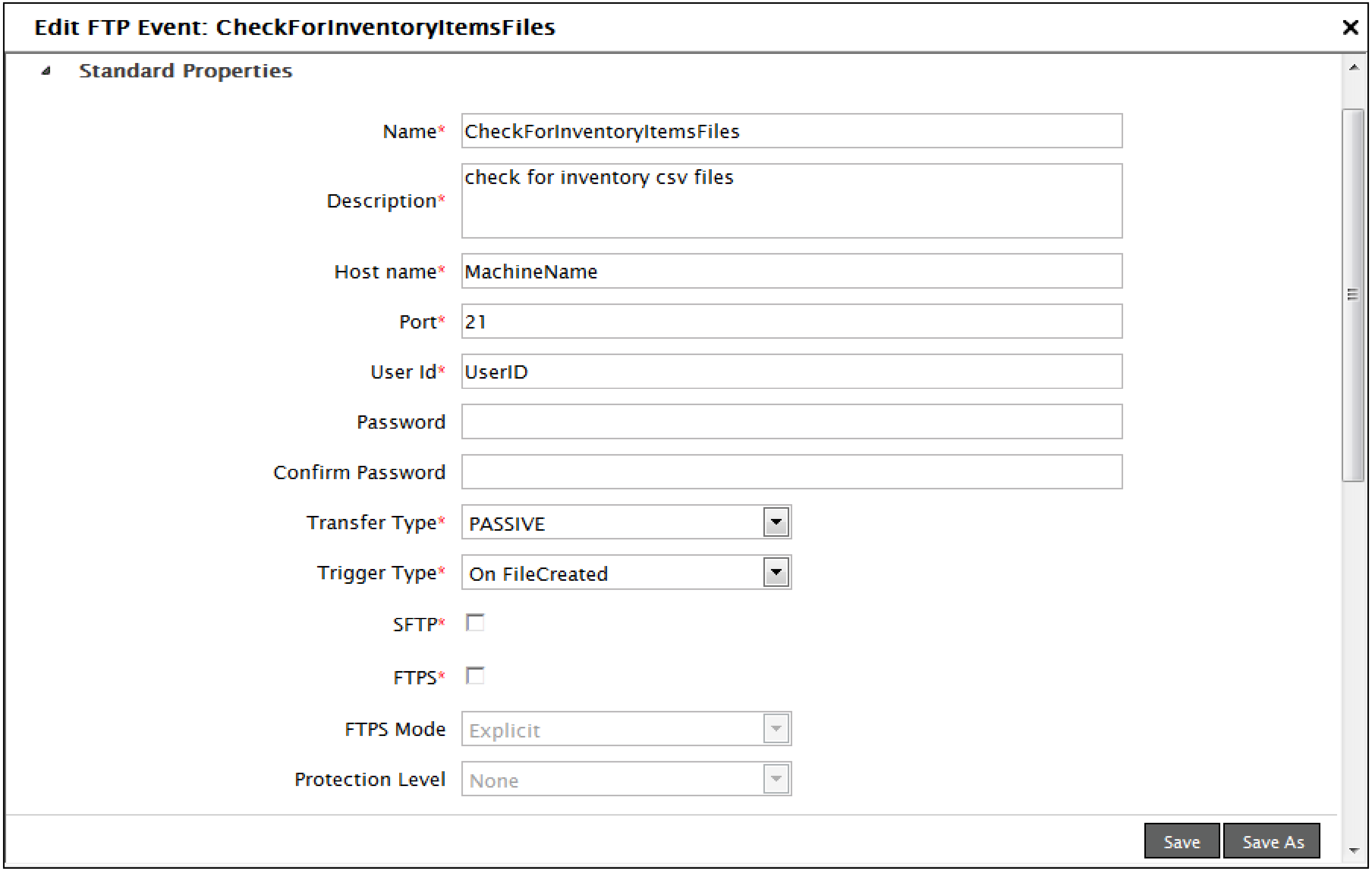


Edit File Event: CheckForPurchaseOrderFiles

Standard Properties

Name*	CheckForPurchaseOrderFiles
Description*	check for new purchase order files
Trigger Type*	On FileCreated
Check for File Modification	□
File Include Criteria*	*.xml
File Exclude Criteria	
File Base Location*	.\Solutions\Demo\PurchaseOrderProcessing\input
Use VFS*	□
Secure	□
User Id*	
Password	
Confirm Password	
Event Start Date	

FTP Event

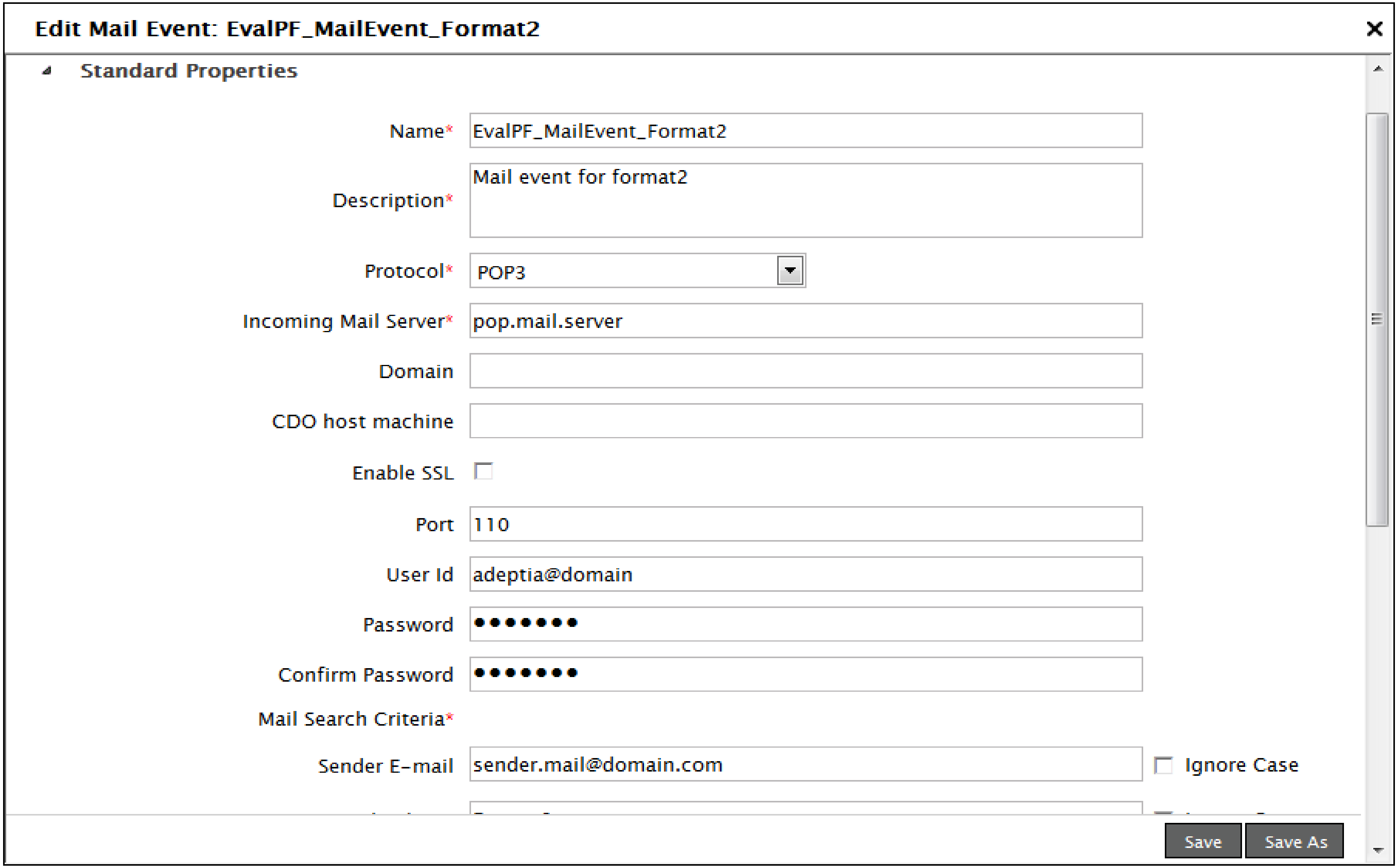


Edit FTP Event: CheckForInventoryItemsFiles

Standard Properties

Name*	CheckForInventoryItemsFiles
Description*	check for inventory csv files
Host name*	MachineName
Port*	21
User Id*	UserID
Password	
Confirm Password	
Transfer Type*	PASSIVE
Trigger Type*	On FileCreated
SFTP*	□
FTPS*	□
FTPS Mode	Explicit
Protection Level	None

Mail Event

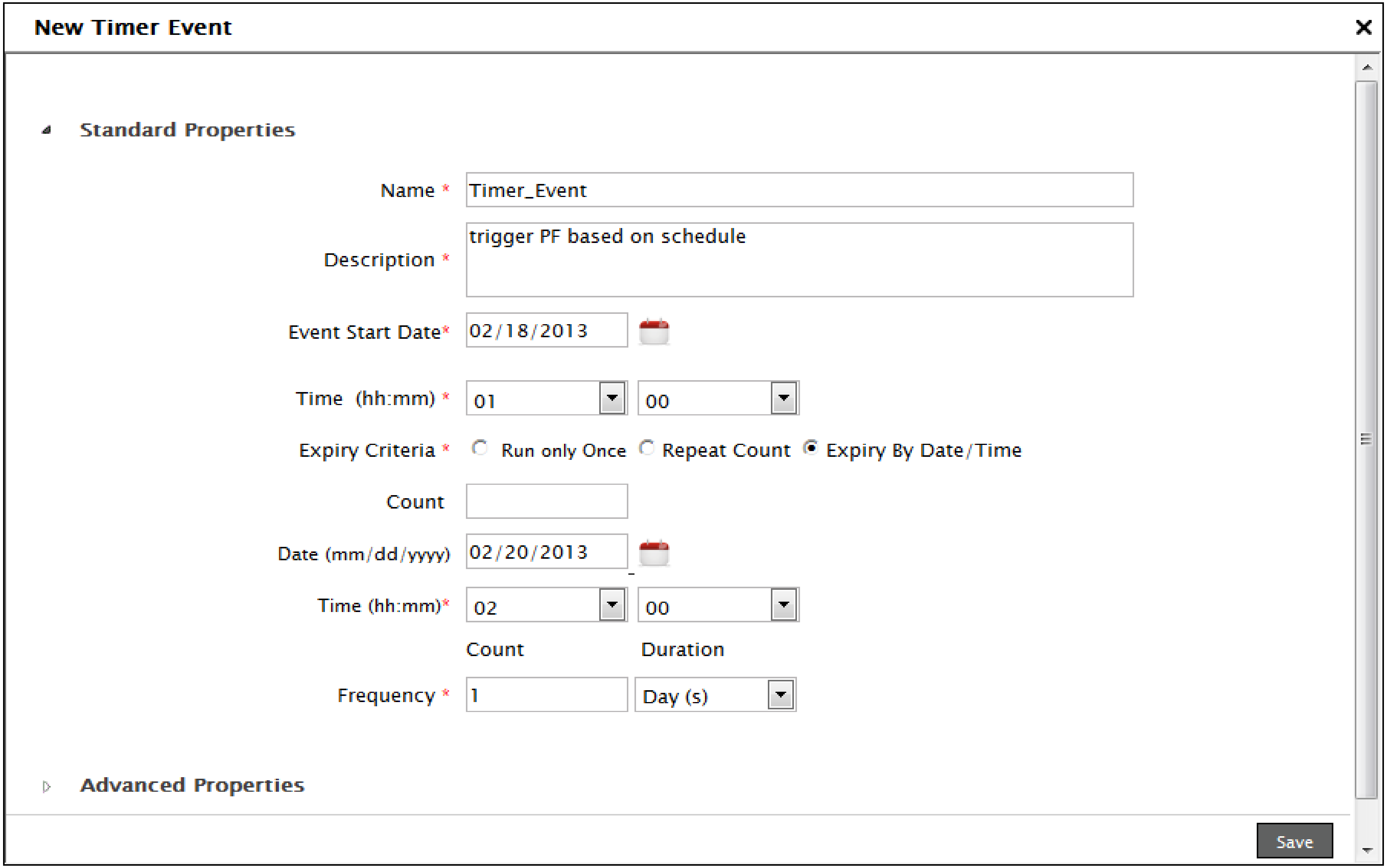


Edit Mail Event: EvalPF_MailEvent_Format2

Standard Properties

Name*	EvalPF_MailEvent_Format2
Description*	Mail event for format2
Protocol*	POP3
Incoming Mail Server*	pop.mail.server
Domain	
CDO host machine	
Enable SSL	□
Port	110
User Id	adeptia@domain
Password	••••••••
Confirm Password	••••••••
Mail Search Criteria*	
Sender E-mail	sender.mail@domain.com □ Ignore Case

Timer Event



New Timer Event

◢ Standard Properties

Name *: Timer_Event

Description *: trigger PF based on schedule

Event Start Date*: 02/18/2013

Time (hh:mm) *: 01 : 00

Expiry Criteria *: ○ Run only Once ○ Repeat Count ● Expiry By Date/Time

Count:

Date (mm/dd/yyyy): 02/20/2013

Time (hh:mm)*: 02 : 00

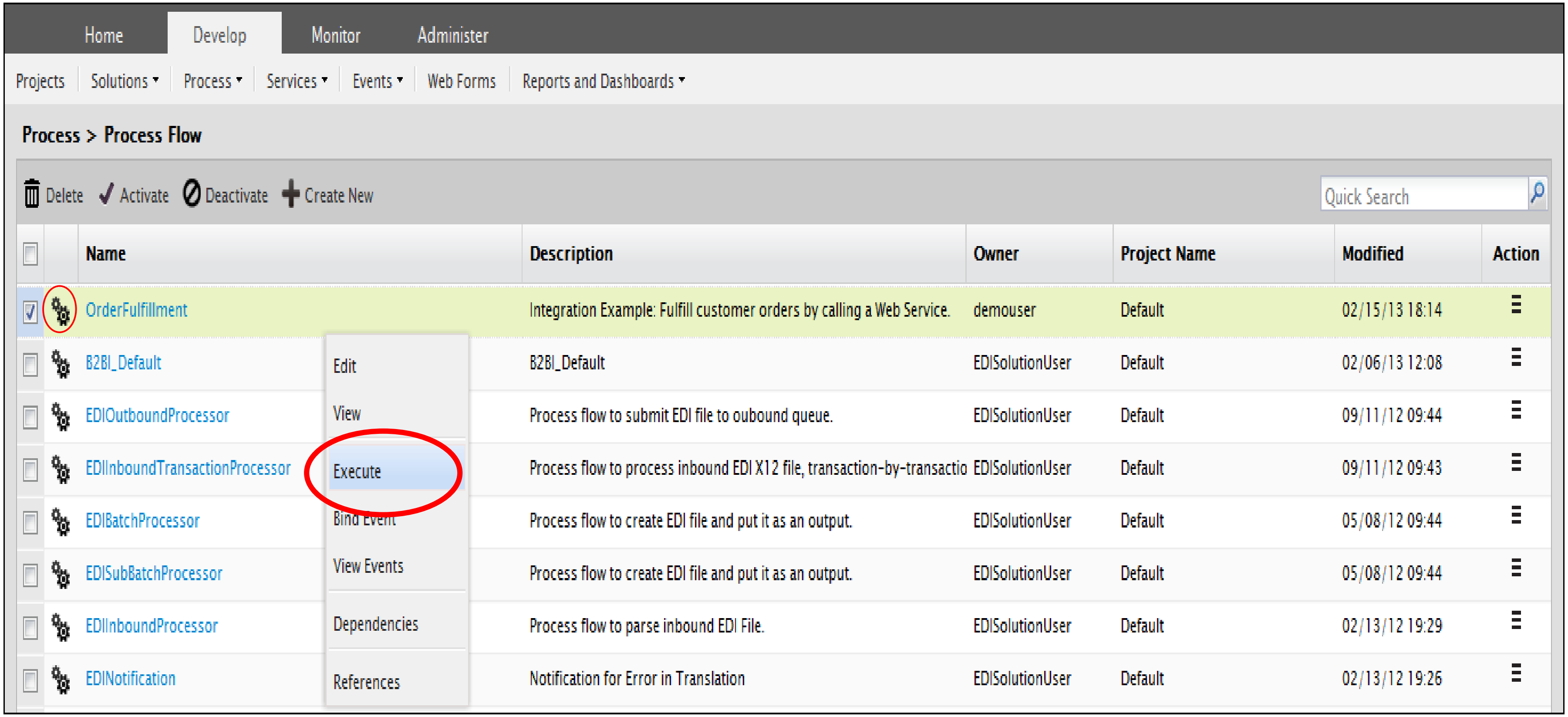
	Count	Duration
Frequency *	1	Day (s)

▷ Advanced Properties

Execute Flow

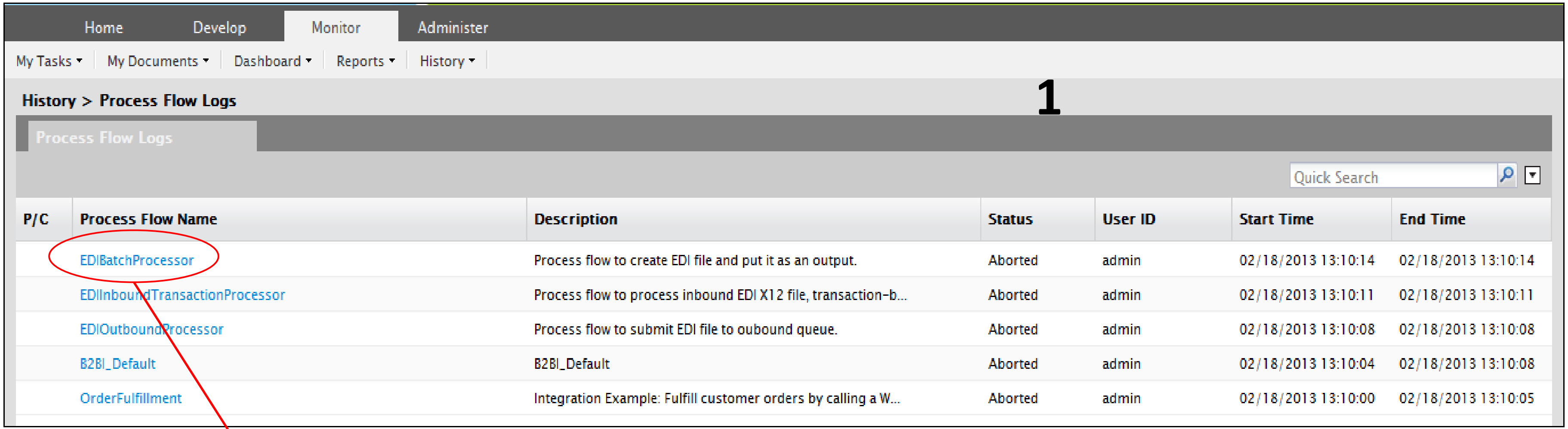
Process Flow Execution

Process Flows can be manually executed or can be triggered by Events.



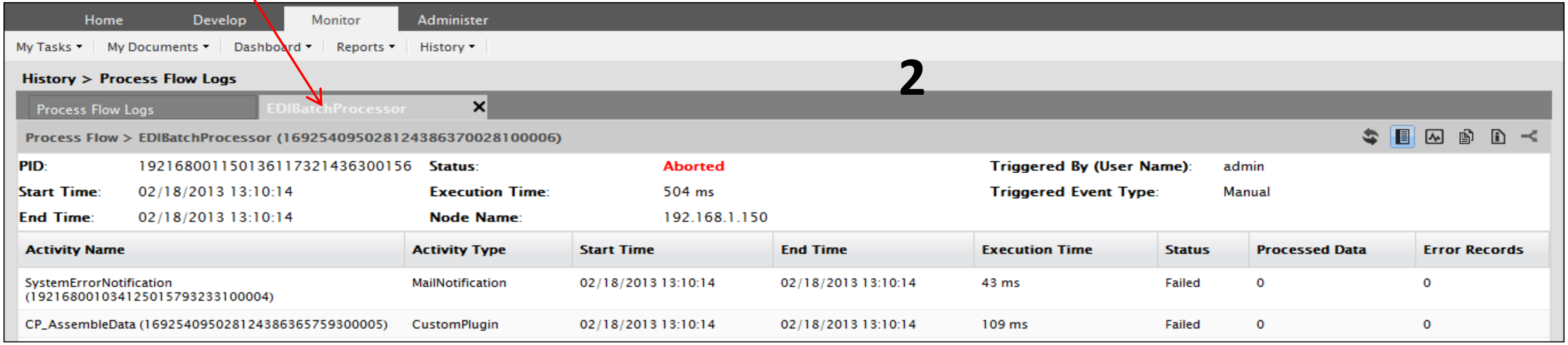
□		Name	Description	Owner	Project Name	Modified	Action
☑		OrderFulfillment	Integration Example: Fulfill customer orders by calling a Web Service.	demouser	Default	02/15/13 18:14	≡
□		B2BI_Default	B2BI_Default	EDISolutionUser	Default	02/06/13 12:08	≡
□		EDIOutboundProcessor	Process flow to submit EDI file to oubound queue.	EDISolutionUser	Default	09/11/12 09:44	≡
□		EDIInboundTransactionProcessor	Process flow to process inbound EDI X12 file, transaction-by-transactio	EDISolutionUser	Default	09/11/12 09:43	≡
□		EDIBatchProcessor	Process flow to create EDI file and put it as an output.	EDISolutionUser	Default	05/08/12 09:44	≡
□		EDISubBatchProcessor	Process flow to create EDI file and put it as an output.	EDISolutionUser	Default	05/08/12 09:44	≡
□		EDIInboundProcessor	Process flow to parse inbound EDI File.	EDISolutionUser	Default	02/13/12 19:29	≡
□		EDINotification	Notification for Error in Translation	EDISolutionUser	Default	02/13/12 19:26	≡

Process Flow Logs



History > Process Flow Logs

P/C	Process Flow Name	Description	Status	User ID	Start Time	End Time
	EDIBatchProcessor	Process flow to create EDI file and put it as an output.	Aborted	admin	02/18/2013 13:10:14	02/18/2013 13:10:14
	EDIInboundTransactionProcessor	Process flow to process inbound EDI X12 file, transaction-b...	Aborted	admin	02/18/2013 13:10:11	02/18/2013 13:10:11
	EDIOutboundProcessor	Process flow to submit EDI file to oubound queue.	Aborted	admin	02/18/2013 13:10:08	02/18/2013 13:10:08
	B2BI_Default	B2BI_Default	Aborted	admin	02/18/2013 13:10:04	02/18/2013 13:10:08
	OrderFulfillment	Integration Example: Fulfill customer orders by calling a W...	Aborted	admin	02/18/2013 13:10:00	02/18/2013 13:10:05



History > Process Flow Logs

Process Flow > EDIBatchProcessor (16925409502812438637002810006)

PID:	19216800115013611732143630015 6	Status:	Aborted	Triggered By (User Name):	admin
Start Time:	02/18/2013 13:10:14	Execution Time:	504 ms	Triggered Event Type:	Manual
End Time:	02/18/2013 13:10:14	Node Name:	192.168.1.150		

Activity Name	Activity Type	Start Time	End Time	Execution Time	Status	Processed Data	Error Records
SystemErrorNotification (19216800103412501579323310004)	MailNotification	02/18/2013 13:10:14	02/18/2013 13:10:14	43 ms	Failed	0	0
CP_AssembleData (16925409502812438636575930005)	CustomPlugin	02/18/2013 13:10:14	02/18/2013 13:10:14	109 ms	Failed	0	0

Training schedule

Day 1

Morning session
9am to 12pm, 10 min break at 10:30am

- Introduction
  - Overview of Adeptia Suite
  - Architecture
  - Project approach
- Process Modeler & Simulation (BPM)
  - Modeler
  - Simulator
- Process Designer
- BPM Exercise – Purchase Requisition Workflow (BPM)
  - Web Forms
  - Workflow

Afternoon session
1pm to 4:30pm, 10 min break @ 2:20pm and 3:20pm

- Integration/ETL Exercises:
  - Employee Benefits
  - Purchase Order
  - Inventory Items
  - Order Fulfillment
- Services: Description and Exercises
  - Sources
  - Targets
  - Data Dictionary
  - Schema
  - Mapping
  - Web Services
  - Workflow
- Events & Triggers

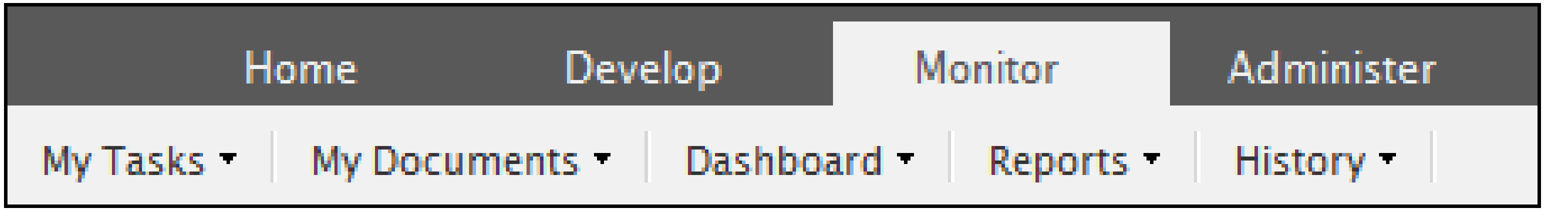
Day 2

Morning session
9am to 12pm, 10 min break at 10:30am

- Advanced Mapping & Process Designer functions
  - Process Designer
    - Decision nodes
    - Context Variables
    - Sub Processes
  - Mapping
    - Math, String and Conditional functions
    - Database lookups
    - Context variables
    - Multiple Source and Target mapping
- Execute flow
  - Trigger
  - Logs
  - Reports

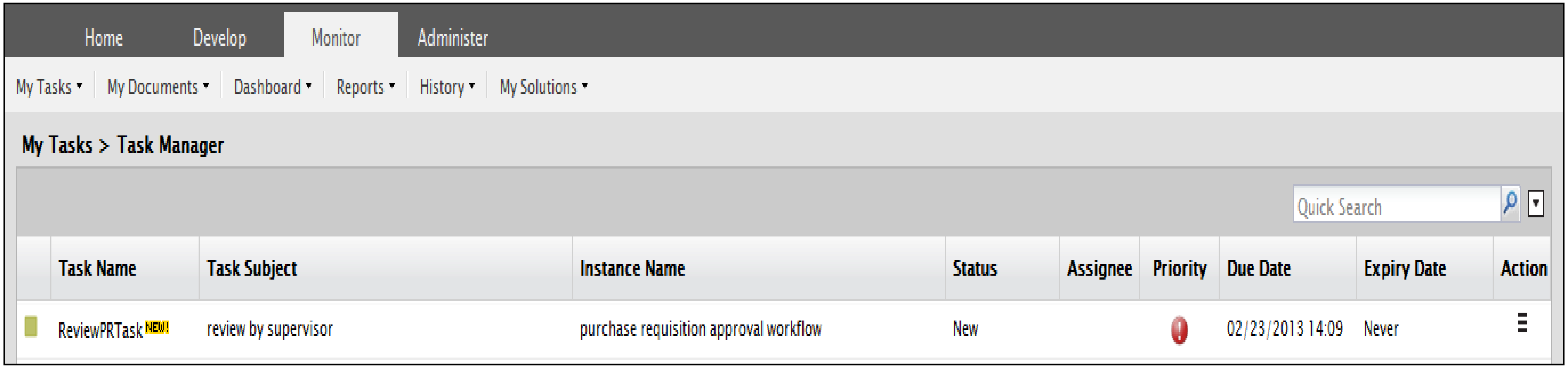
Afternoon session
1pm to 4pm, 10 min break @ 2:30pm

- Managing Solutions
  - My Tasks and Task History
  - Solution and Performance Monitoring
  - Logs
- Administration
  - User management
  - Configuration and Properties
  - System monitoring
  - Maintenance activities



Managing Tasks and Solutions

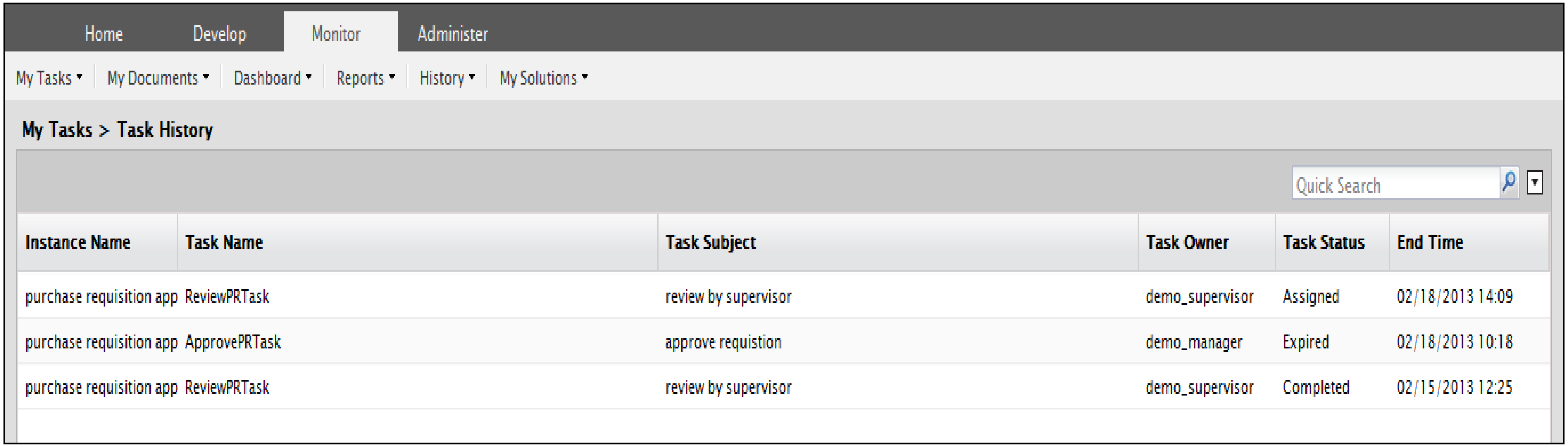
My Tasks > Task Manager



My Tasks > Task Manager

	Task Name	Task Subject	Instance Name	Status	Assignee	Priority	Due Date	Expiry Date	Action
🟩	ReviewPRTask NEW!	review by supervisor	purchase requisition approval workflow	New		!	02/23/2013 14:09	Never	☰

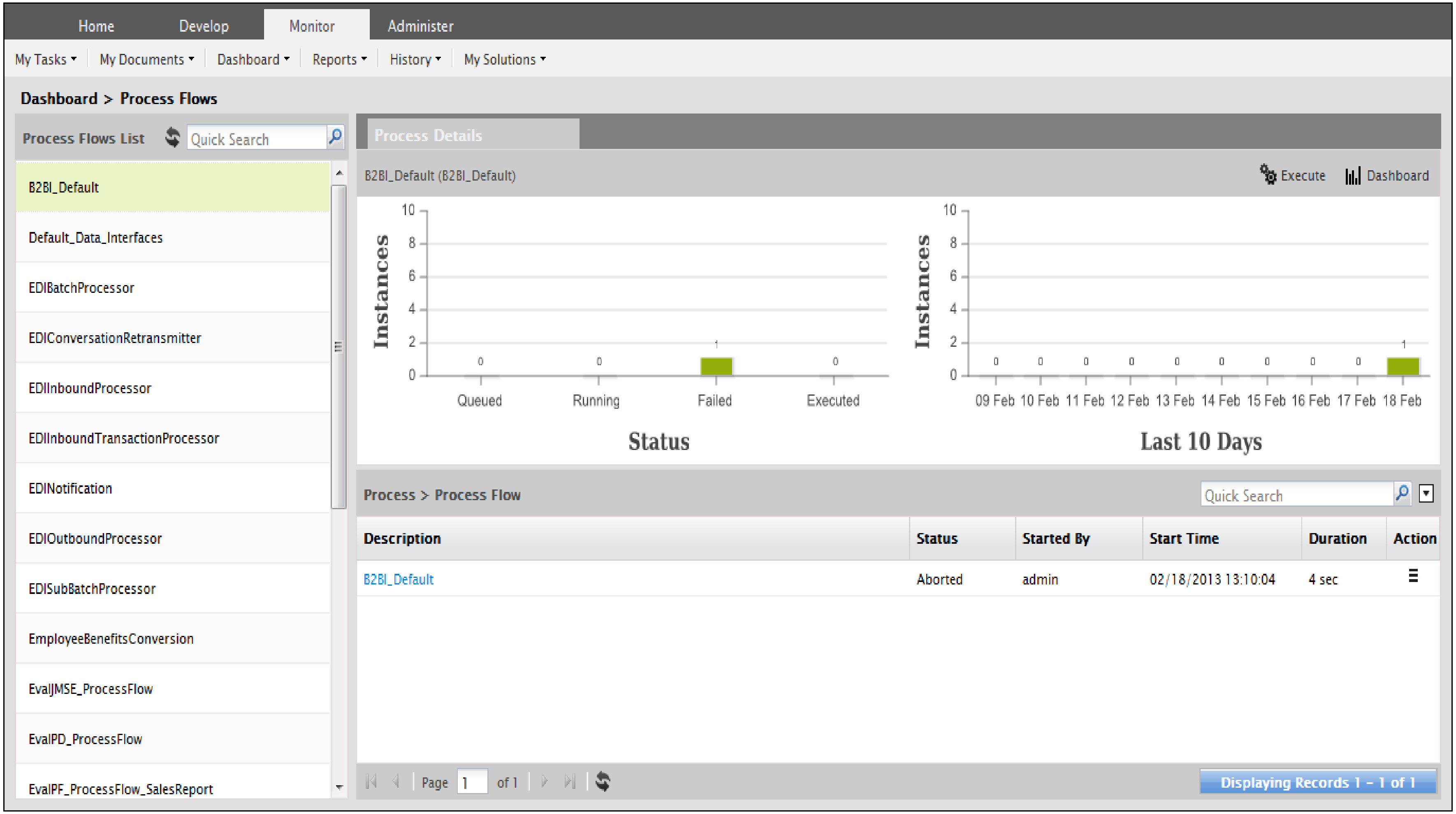
My Tasks > Task History



My Tasks > Task History

Instance Name	Task Name	Task Subject	Task Owner	Task Status	End Time
purchase requisition app	ReviewPRTask	review by supervisor	demo_supervisor	Assigned	02/18/2013 14:09
purchase requisition app	ApprovePRTask	approve requistion	demo_manager	Expired	02/18/2013 10:18
purchase requisition app	ReviewPRTask	review by supervisor	demo_supervisor	Completed	02/15/2013 12:25

Monitor > Dashboard



Dashboard > Process Flows

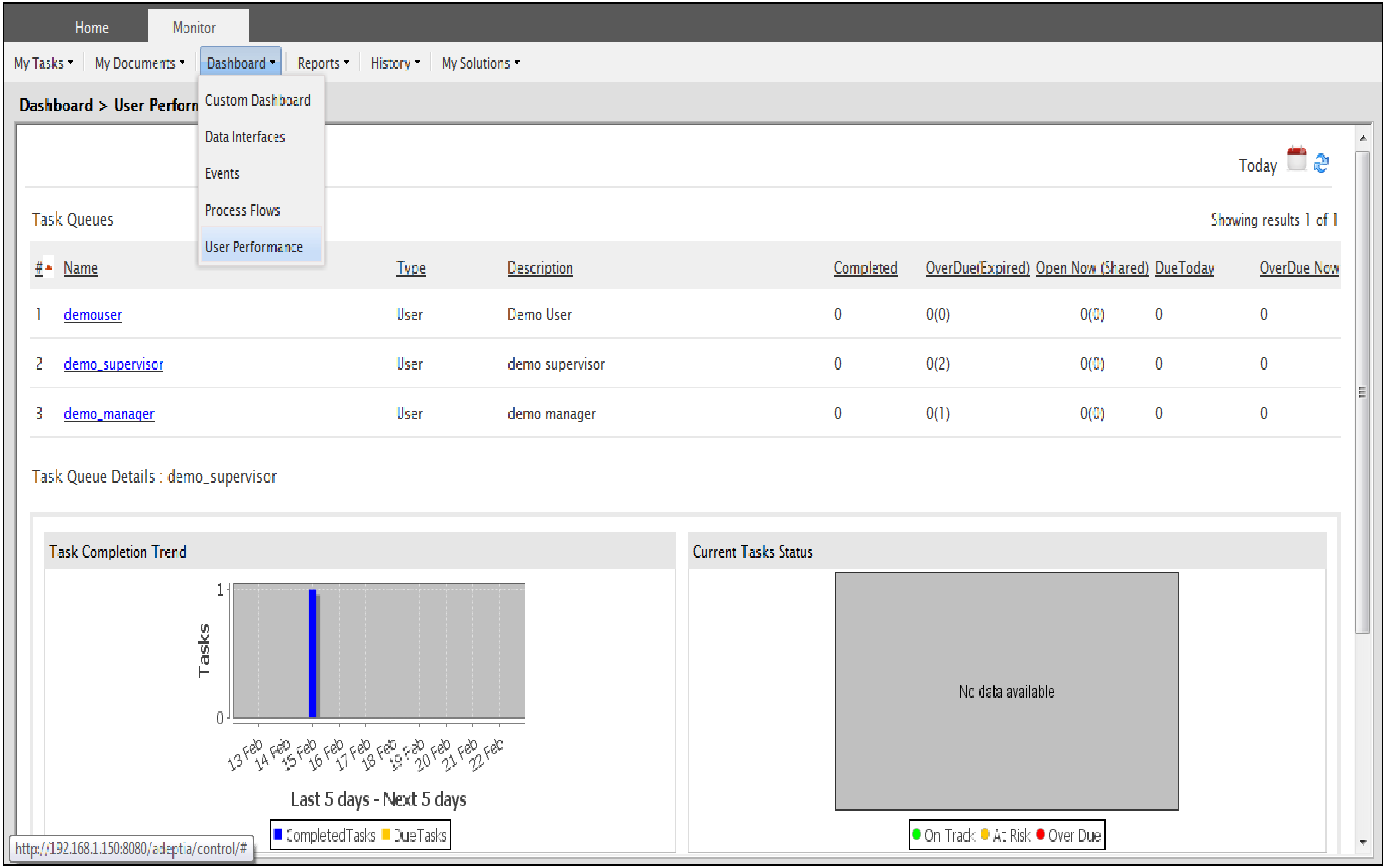
- B2BI_Default
- Default_Data_Interfaces
- EDIBatchProcessor
- EDIConversationRetransmitter
- EDIInboundProcessor
- EDIInboundTransactionProcessor
- EDINotification
- EDIOutboundProcessor
- EDISubBatchProcessor
- EmployeeBenefitsConversion
- EvalJMSE_ProcessFlow
- EvalPD_ProcessFlow
- EvalPF_ProcessFlow_SalesReport

B2BI_Default (B2BI_Default)

Process > Process Flow

Description	Status	Started By	Start Time	Duration	Action
B2BI_Default	Aborted	admin	02/18/2013 13:10:04	4 sec	☰

Monitor > Dashboard > User Performance



Dashboard > User Perforn

Task Queues

Showing results 1 of 1

#	Name	Type	Description	Completed	OverDue(Expired)	Open Now (Shared)	DueToday	OverDue Now
1	demouser	User	Demo User	0	0(0)	0(0)	0	0
2	demo_supervisor	User	demo supervisor	0	0(2)	0(0)	0	0
3	demo_manager	User	demo manager	0	0(1)	0(0)	0	0

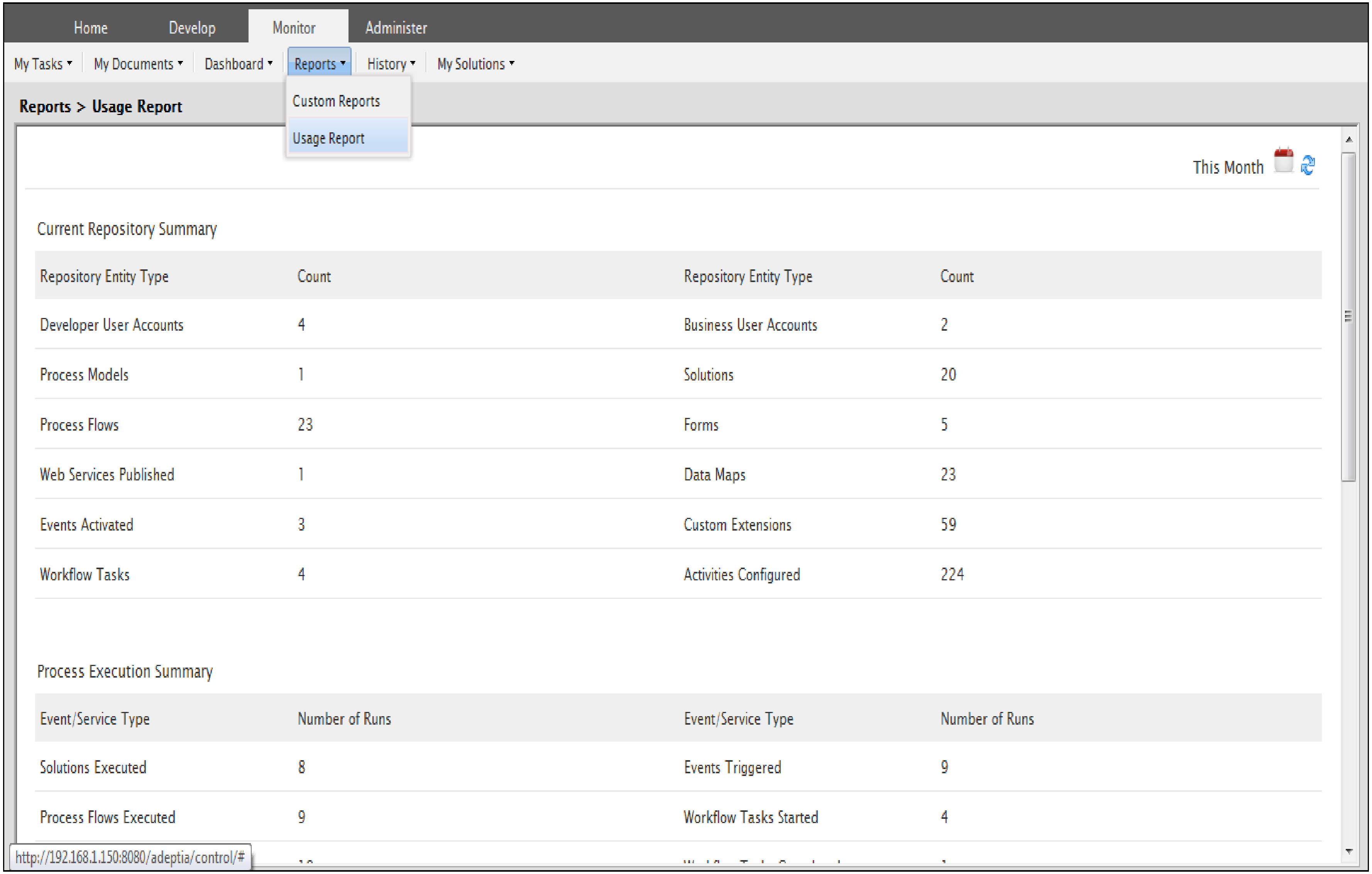
Task Queue Details : demo_supervisor

Task Completion Trend

Current Tasks Status

No data available
● On Track ● At Risk ● Over Due
http://192.168.1.150:8080/adeptia/control/#

My Monitor > Usage Report



Reports > Usage Report

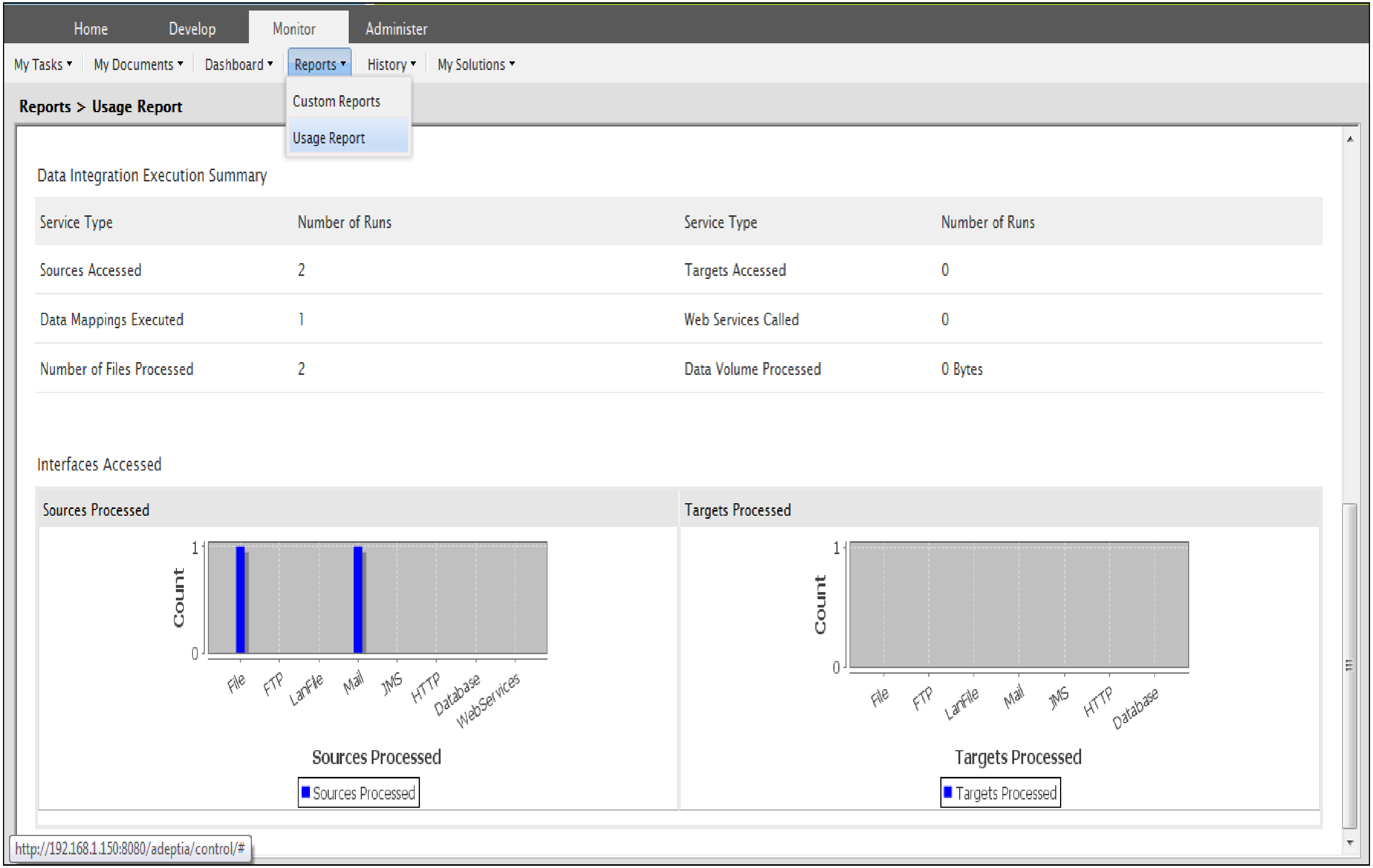
Current Repository Summary

Repository Entity Type	Count	Repository Entity Type	Count
Developer User Accounts	4	Business User Accounts	2
Process Models	1	Solutions	20
Process Flows	23	Forms	5
Web Services Published	1	Data Maps	23
Events Activated	3	Custom Extensions	59
Workflow Tasks	4	Activities Configured	224

Process Execution Summary

Event/Service Type	Number of Runs	Event/Service Type	Number of Runs
Solutions Executed	8	Events Triggered	9
Process Flows Executed	9	Workflow Tasks Started	4

My Monitor > Usage Report



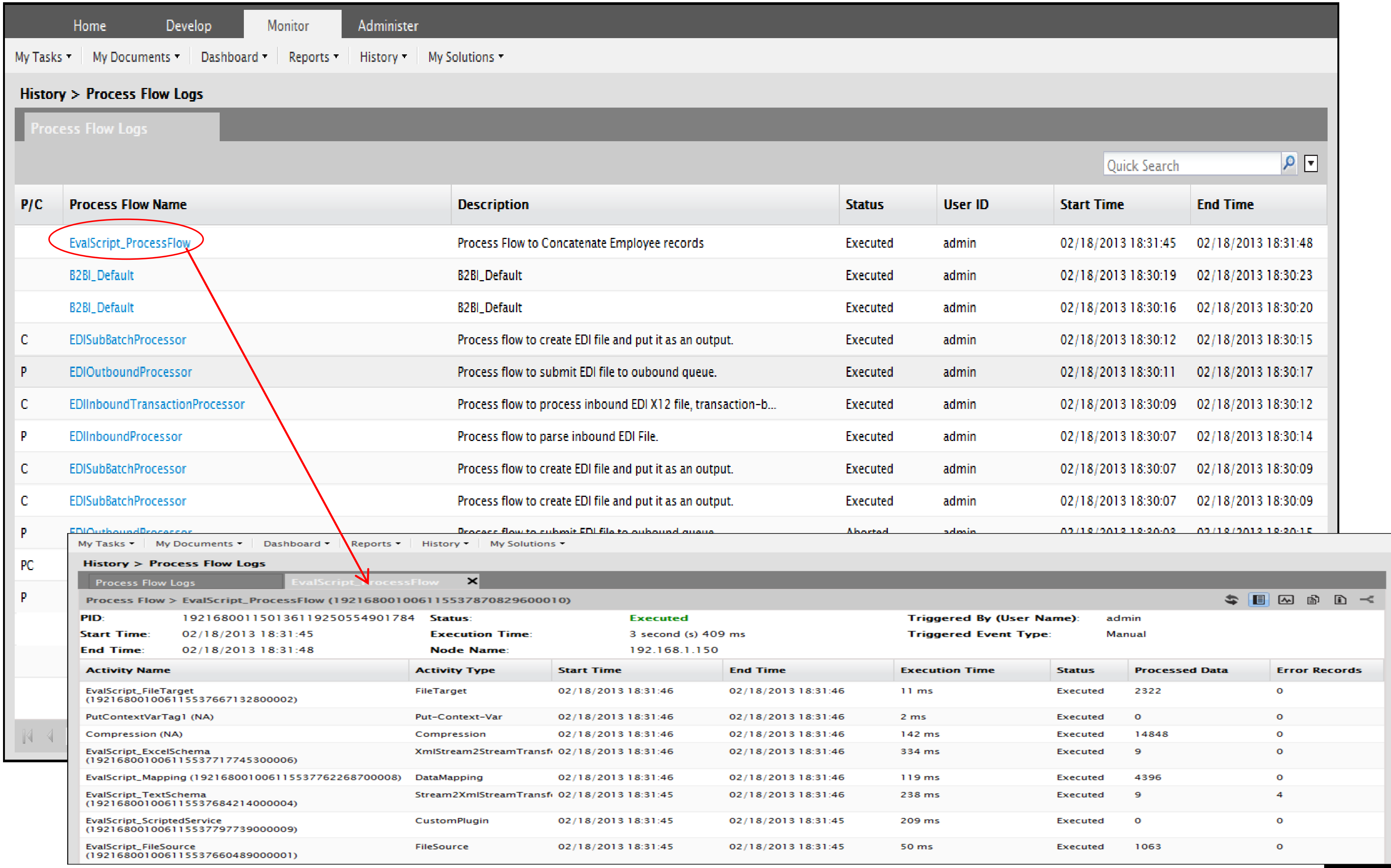
Reports > Usage Report

Data Integration Execution Summary

Service Type	Number of Runs	Service Type	Number of Runs
Sources Accessed	2	Targets Accessed	0
Data Mappings Executed	1	Web Services Called	0
Number of Files Processed	2	Data Volume Processed	0 Bytes

Interfaces Accessed

History > Process Flow Logs



History > Process Flow Logs

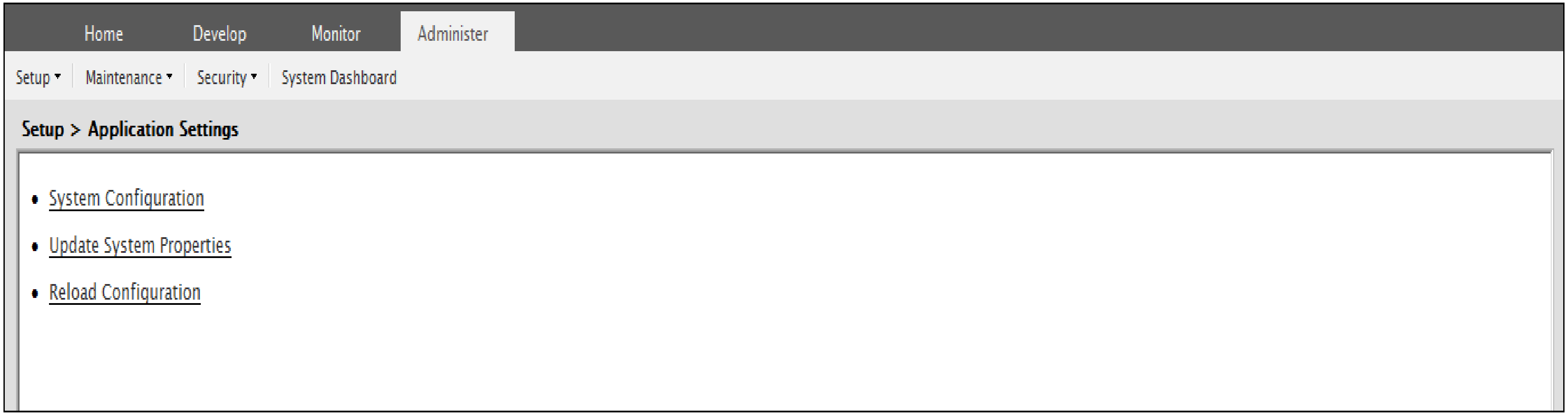
P/C	Process Flow Name	Description	Status	User ID	Start Time	End Time
	EvalScript_ProcessFlow	Process Flow to Concatenate Employee records	Executed	admin	02/18/2013 18:31:45	02/18/2013 18:31:48
	B2BI_Default	B2BI_Default	Executed	admin	02/18/2013 18:30:19	02/18/2013 18:30:23
	B2BI_Default	B2BI_Default	Executed	admin	02/18/2013 18:30:16	02/18/2013 18:30:20
C	EDISubBatchProcessor	Process flow to create EDI file and put it as an output.	Executed	admin	02/18/2013 18:30:12	02/18/2013 18:30:15
P	EDIOutboundProcessor	Process flow to submit EDI file to oubound queue.	Executed	admin	02/18/2013 18:30:11	02/18/2013 18:30:17
C	EDIInboundTransactionProcessor	Process flow to process inbound EDI X12 file, transaction-b...	Executed	admin	02/18/2013 18:30:09	02/18/2013 18:30:12
P	EDIInboundProcessor	Process flow to parse inbound EDI File.	Executed	admin	02/18/2013 18:30:07	02/18/2013 18:30:14
C	EDISubBatchProcessor	Process flow to create EDI file and put it as an output.	Executed	admin	02/18/2013 18:30:07	02/18/2013 18:30:09
C	EDISubBatchProcessor	Process flow to create EDI file and put it as an output.	Executed	admin	02/18/2013 18:30:07	02/18/2013 18:30:09
P	EDIOutboundProcessor	Process flow to submit EDI file to oubound queue.	Aborted	admin	02/18/2013 18:30:03	02/18/2013 18:30:15
PC						
P						

History > Process Flow Logs

Process Flow > EvalScript_ProcessFlow (192168001006115537870829600010)

PID:	192168001150136119250554901784	Status:	Executed	Triggered By (User Name):	admin
Start Time:	02/18/2013 18:31:45	Execution Time:	3 second (s) 409 ms	Triggered Event Type:	Manual
End Time:	02/18/2013 18:31:48	Node Name:	192.168.1.150		

Activity Name	Activity Type	Start Time	End Time	Execution Time	Status	Processed Data	Error Records
EvalScript_FileTarget (192168001006115537667132800002)	FileTarget	02/18/2013 18:31:46	02/18/2013 18:31:46	11 ms	Executed	2322	0
PutContextVarTag1 (NA)	Put-Context-Var	02/18/2013 18:31:46	02/18/2013 18:31:46	2 ms	Executed	0	0
Compression (NA)	Compression	02/18/2013 18:31:46	02/18/2013 18:31:46	142 ms	Executed	14848	0
EvalScript_ExcelSchema (192168001006115537717745300006)	XmlStream2StreamTransf	02/18/2013 18:31:46	02/18/2013 18:31:46	334 ms	Executed	9	0
EvalScript_Mapping (192168001006115537762268700008)	DataMapping	02/18/2013 18:31:46	02/18/2013 18:31:46	119 ms	Executed	4396	0
EvalScript_TextSchema (192168001006115537684214000004)	Stream2XmlStreamTransf	02/18/2013 18:31:45	02/18/2013 18:31:46	238 ms	Executed	9	4
EvalScript_ScriptedService (192168001006115537797739000009)	CustomPlugin	02/18/2013 18:31:45	02/18/2013 18:31:45	209 ms	Executed	0	0
EvalScript_FileSource (192168001006115537660489000001)	FileSource	02/18/2013 18:31:45	02/18/2013 18:31:45	50 ms	Executed	1063	0



Setup > Application Settings

- System Configuration
- Update System Properties
- Reload Configuration

Administration

Administer

Allows users to manage :

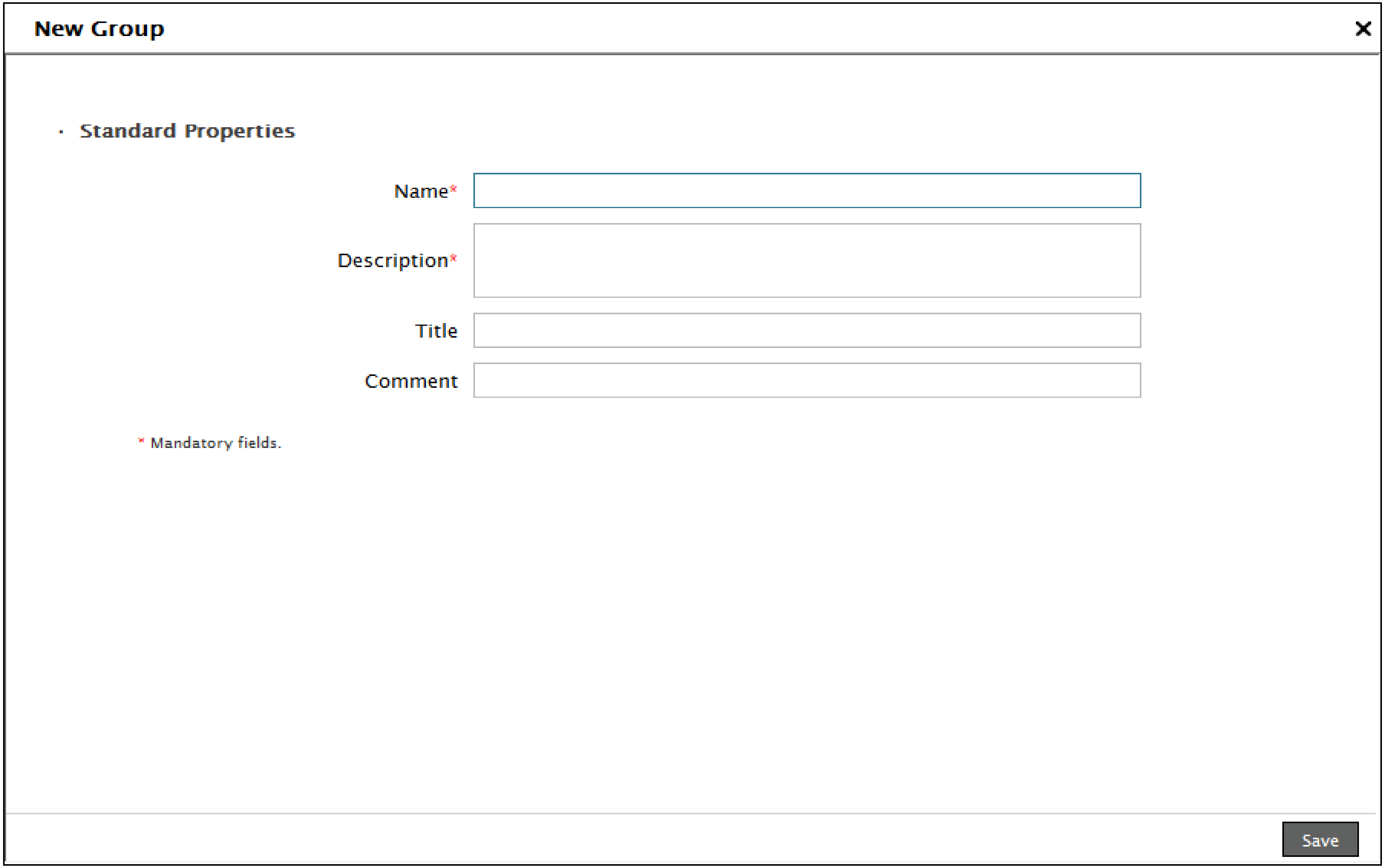
- Groups
- Users
- Application Settings
- Configure Scheduler
- Clustering and Database Connectors etc**.**

Administer - Configure

This includes administration of user and groups, system properties and scheduler.

- Group
- User
- Application Settings
- Start/Stop Scheduler

User Groups



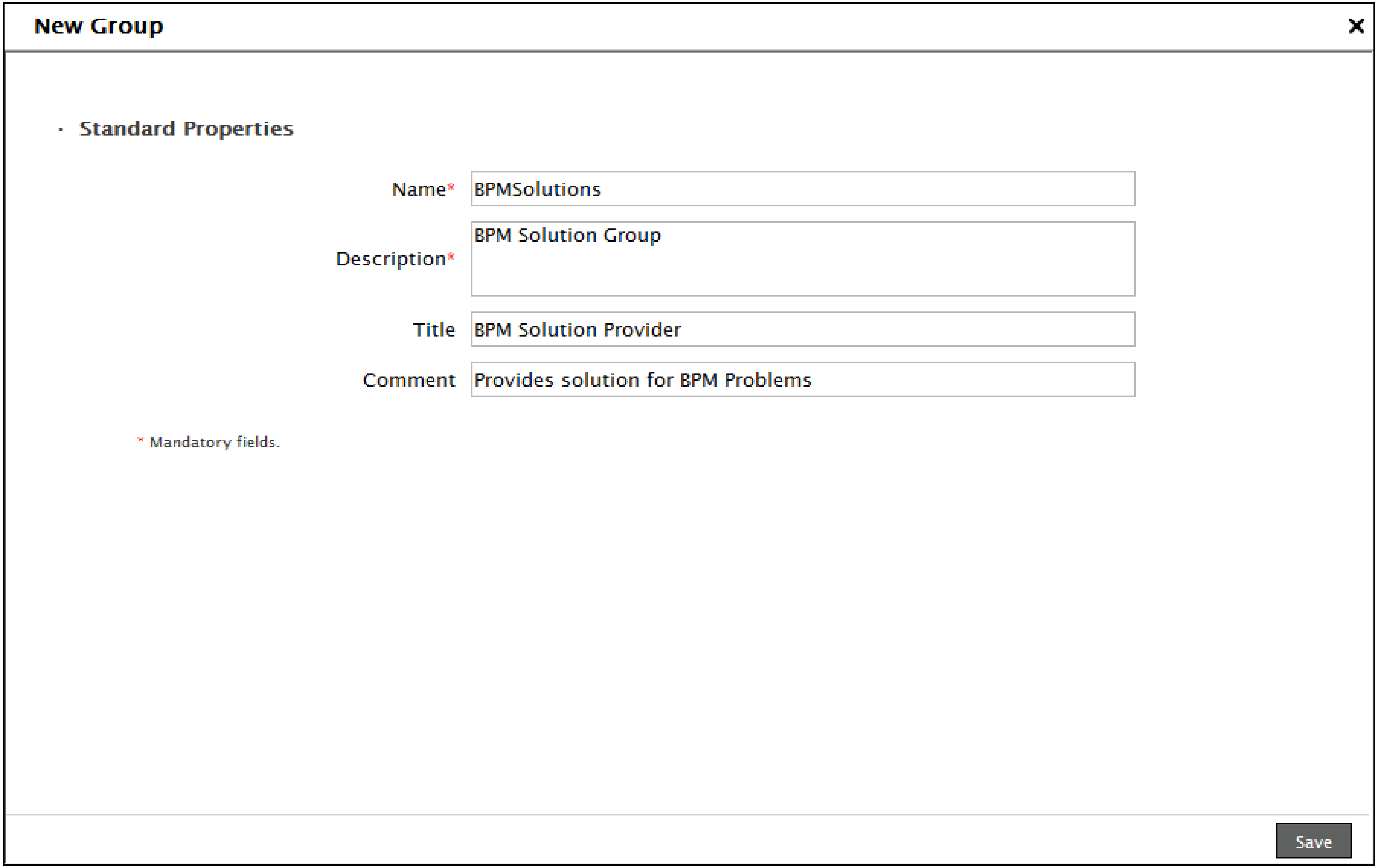
New Group

- Standard Properties

Name*	
Description*	
Title	
Comment	

* Mandatory fields.

User Group - Exercise



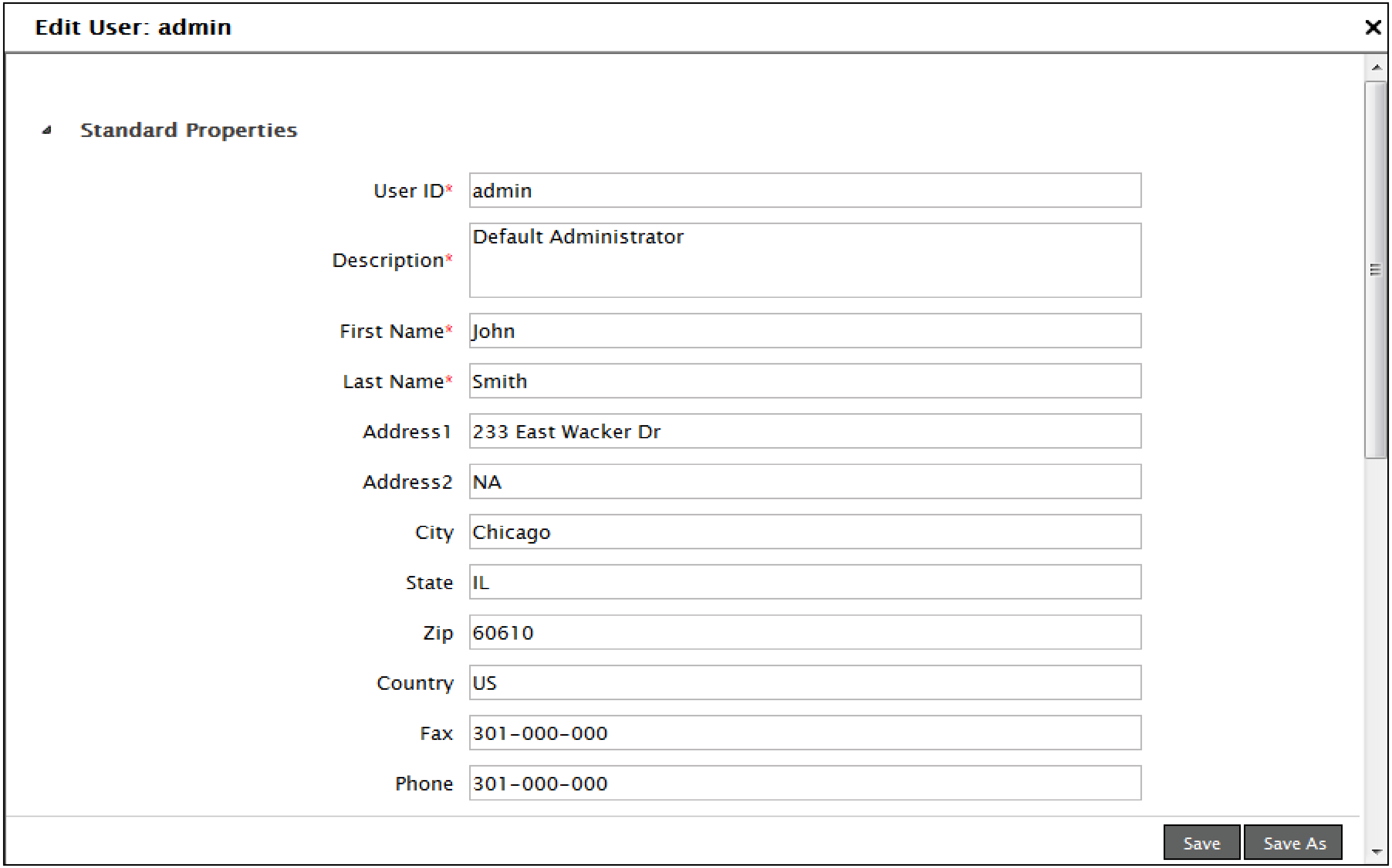
New Group

- Standard Properties

Name*	BPMSolutions
Description*	BPM Solution Group
Title	BPM Solution Provider
Comment	Provides solution for BPM Problems

* Mandatory fields.

User

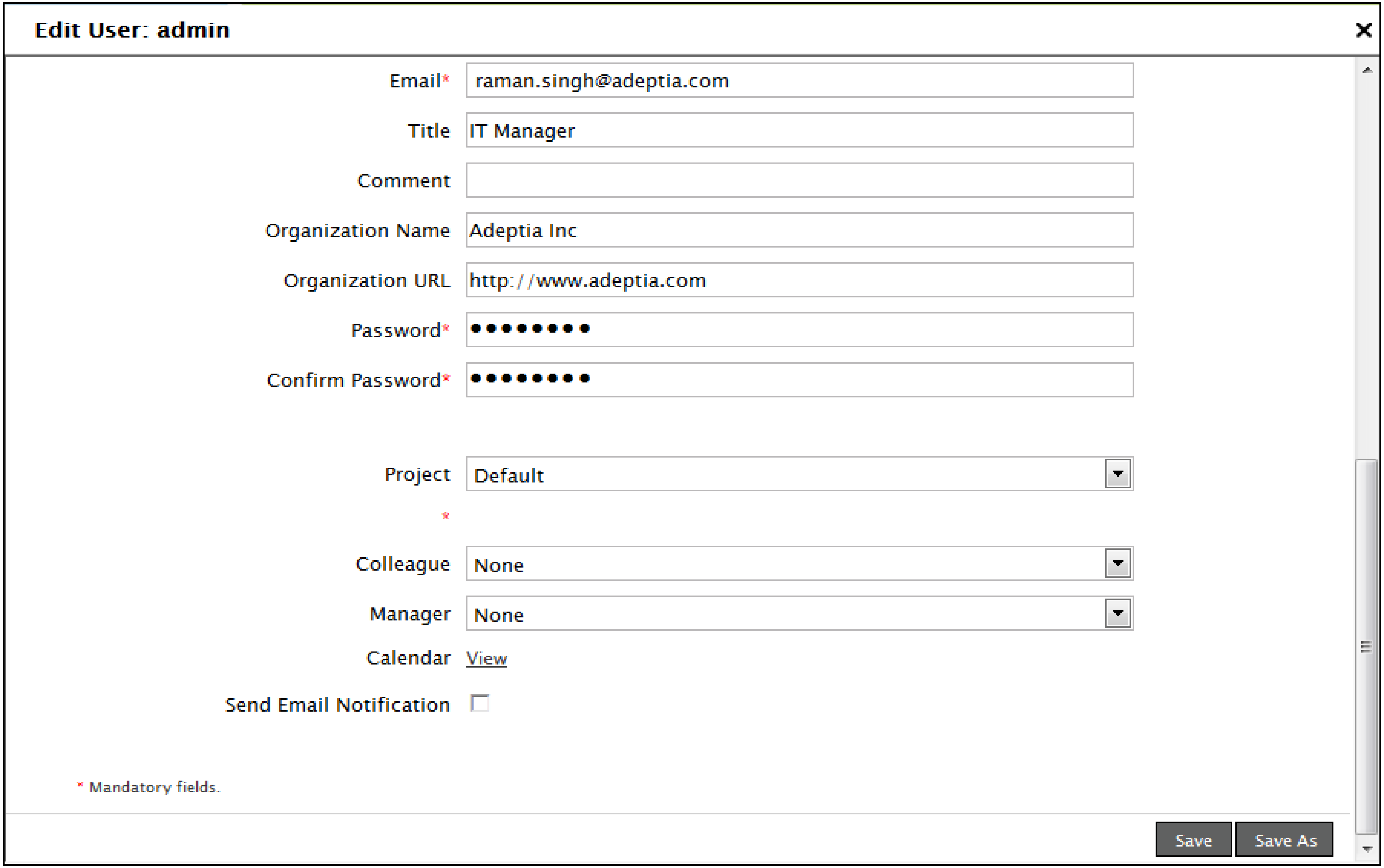


Edit User: admin

Standard Properties

User ID*	admin
Description*	Default Administrator
First Name*	John
Last Name*	Smith
Address1	233 East Wacker Dr
Address2	NA
City	Chicago
State	IL
Zip	60610
Country	US
Fax	301-000-000
Phone	301-000-000

User – Cont.

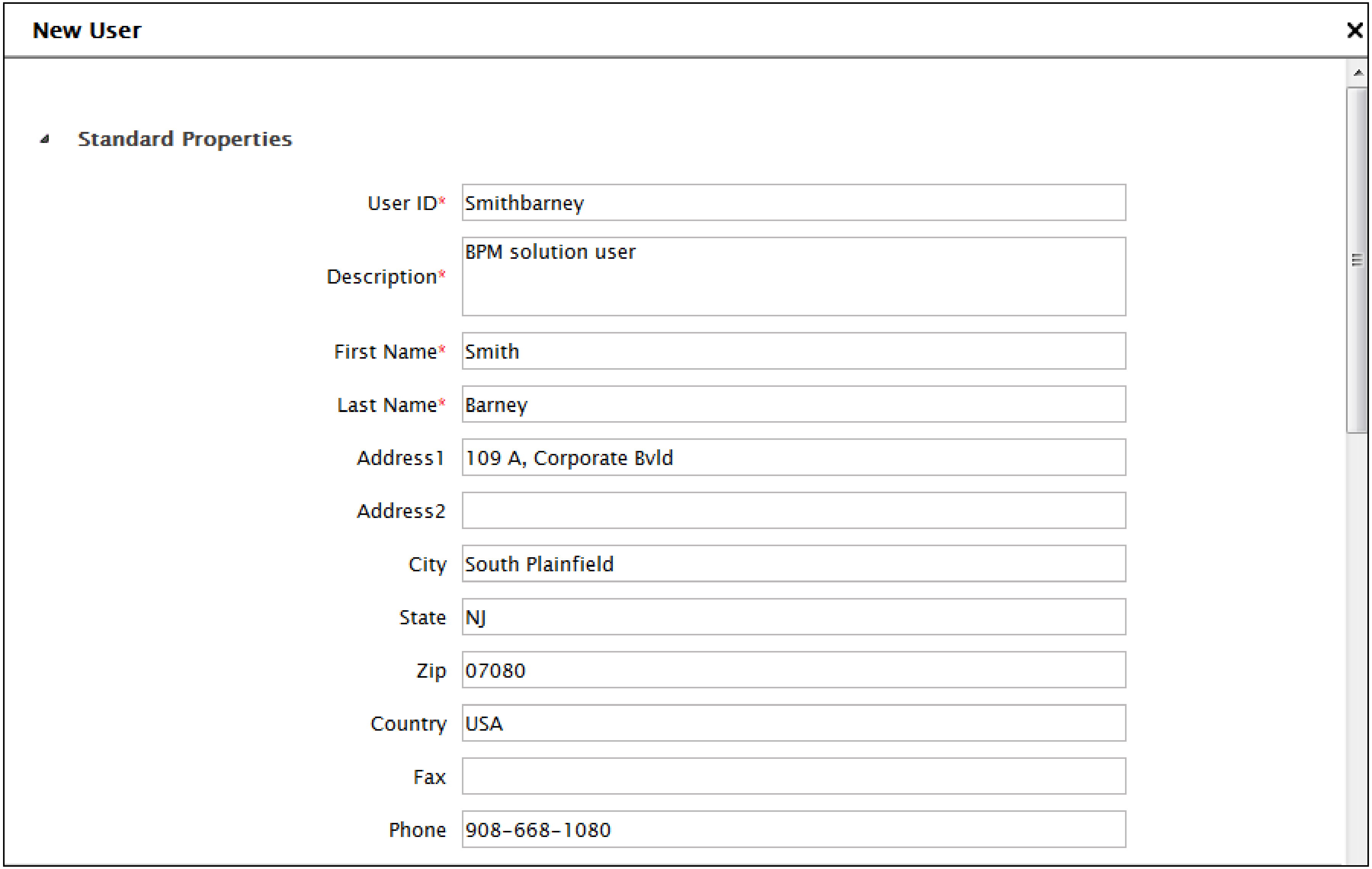


Edit User: admin

Email*	raman.singh@adeptia.com
Title	IT Manager
Comment	
Organization Name	Adeptia Inc
Organization URL	http://www.adeptia.com
Password*	●●●●●●●●●
Confirm Password*	●●●●●●●●●
Project	Default
*	
Colleague	None
Manager	None
Calendar	View
Send Email Notification	□

* Mandatory fields.

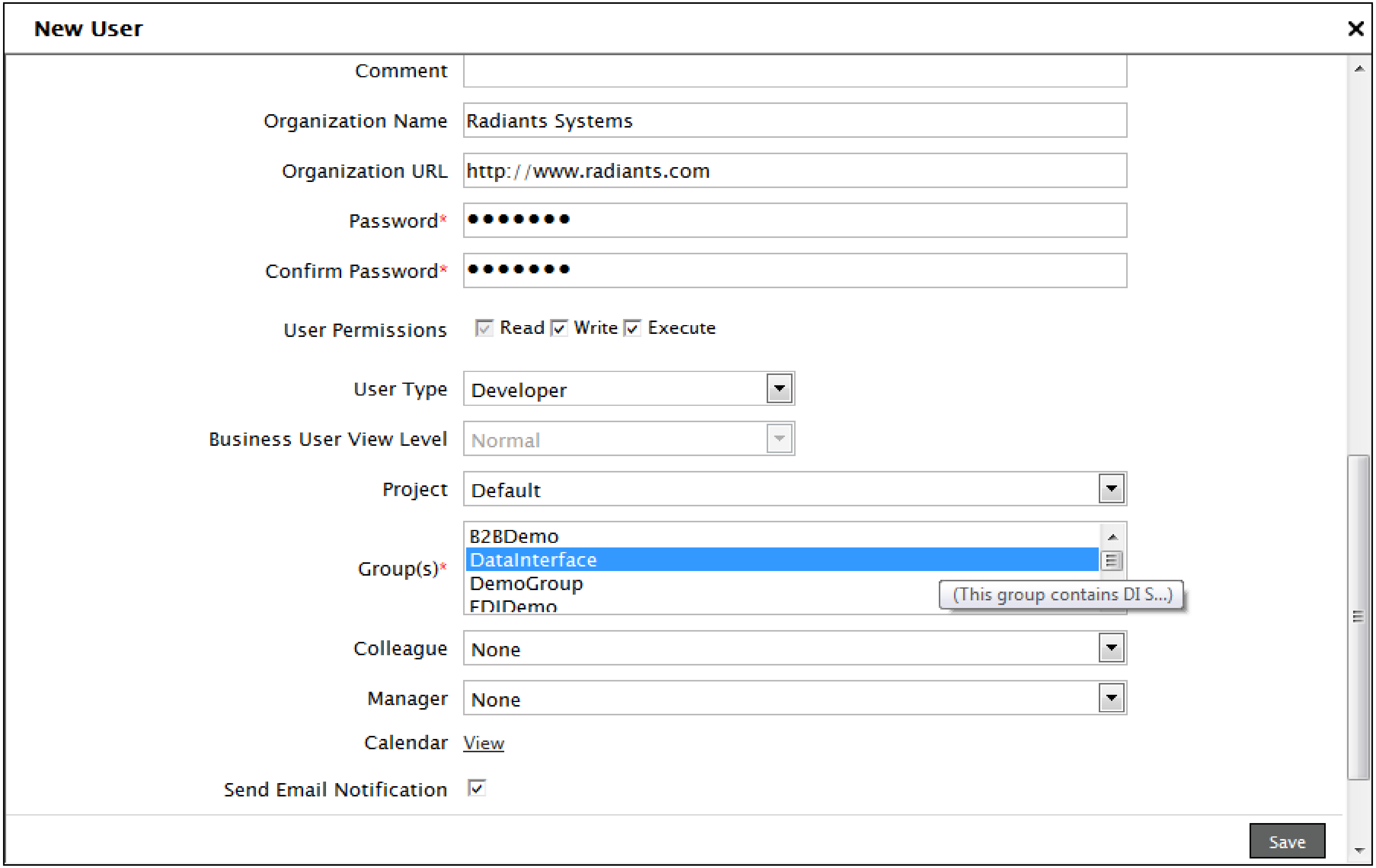
Administer – User - Exercise



◢ Standard Properties

User ID*	Smithbarney
Description*	BPM solution user
First Name*	Smith
Last Name*	Barney
Address1	109 A, Corporate Bvld
Address2	
City	South Plainfield
State	NJ
Zip	07080
Country	USA
Fax	
Phone	908-668-1080

Administer – User – Exercise Cont.

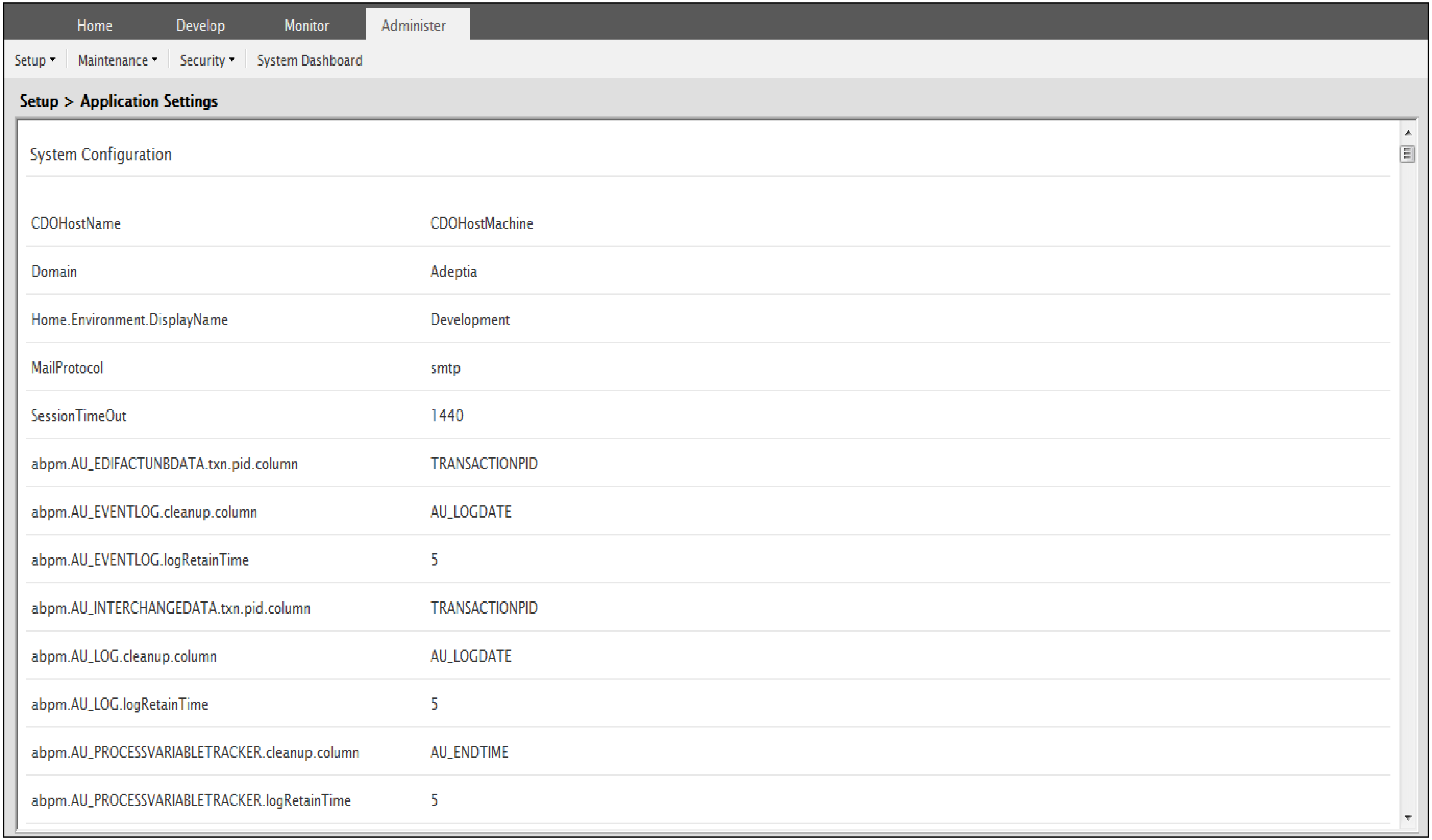


New User

Comment	
Organization Name	Radiants Systems
Organization URL	http://www.radiants.com
Password*	••••••••
Confirm Password*	••••••••
User Permissions	☑ Read ☑ Write ☑ Execute
User Type	Developer
Business User View Level	Normal
Project	Default
Group(s)*	B2BDemo DataInterface DemoGroup EDIDemo
Colleague	None
Manager	None
Calendar	View
Send Email Notification	☑

Application Settings

Allows to view System Configuration and update Adeptia Suite property file



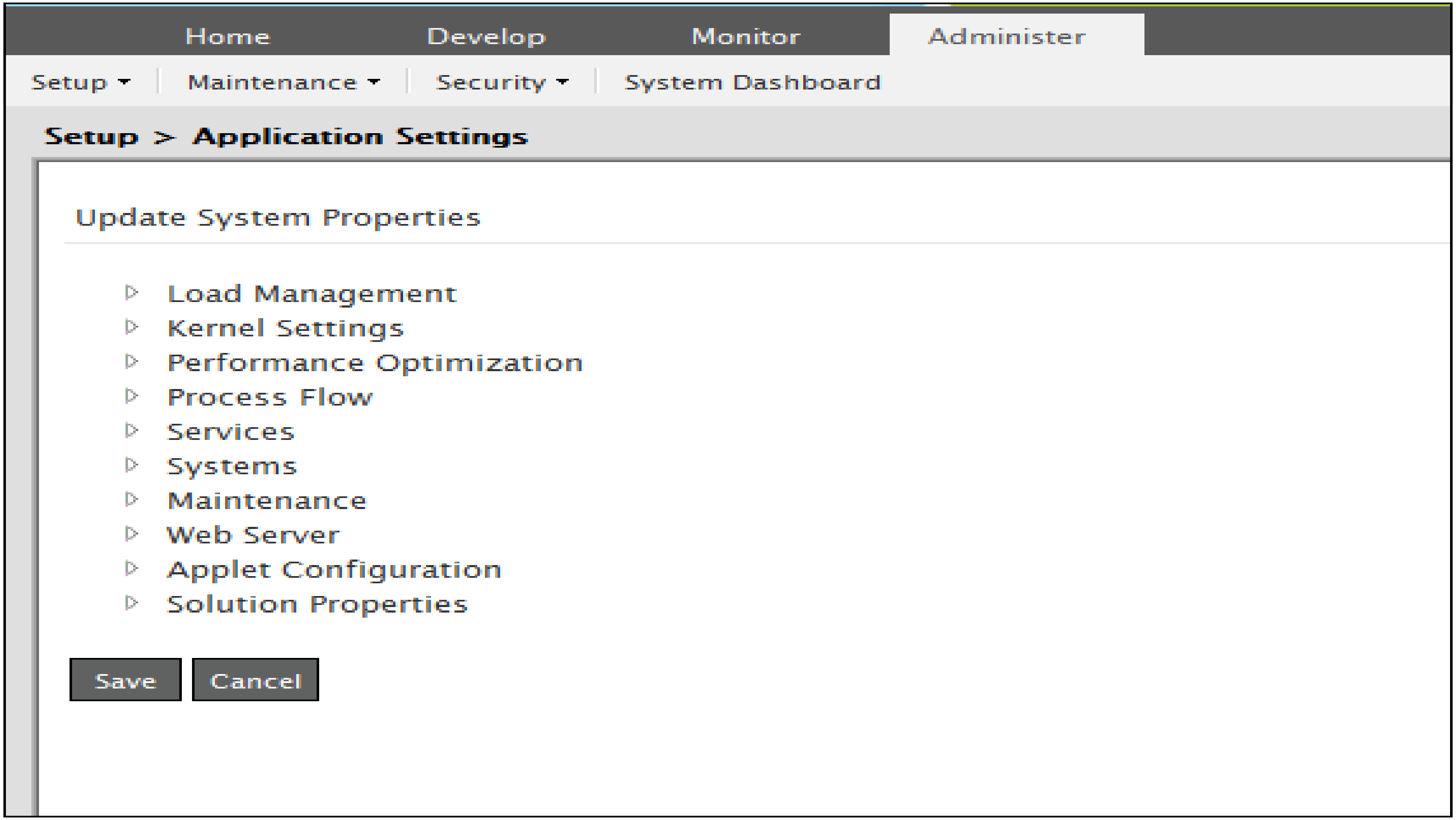
Setup > Application Settings

System Configuration

CDOHostName	CDOHostMachine
Domain	Adeptia
Home.Environment.DisplayName	Development
MailProtocol	smtp
SessionTimeOut	1440
abpm.AU_EDIFACTUNBDATA.txn.pid.column	TRANSACTIONPID
abpm.AU_EVENTLOG.cleanup.column	AU_LOGDATE
abpm.AU_EVENTLOG.logRetainTime	5
abpm.AU_INTERCHANGEDATA.txn.pid.column	TRANSACTIONPID
abpm.AU_LOG.cleanup.column	AU_LOGDATE
abpm.AU_LOG.logRetainTime	5
abpm.AU_PROCESSVARIABLETRACKER.cleanup.column	AU_ENDTIME
abpm.AU_PROCESSVARIABLETRACKER.logRetainTime	5

Update Adeptia Suite Settings

Update Adeptia Suite property file

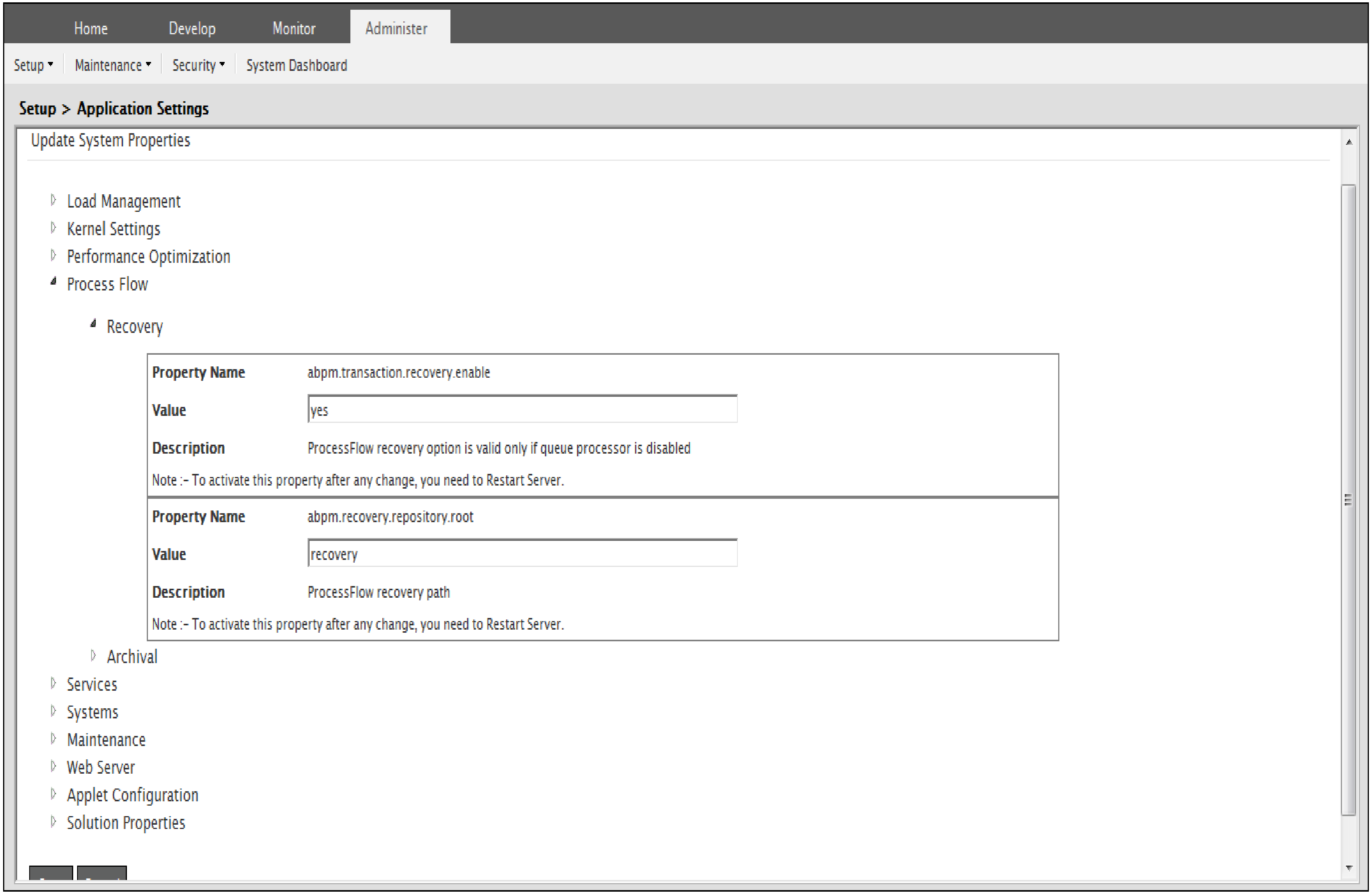


Setup > Application Settings

Update System Properties

- ▷ Load Management
- ▷ Kernel Settings
- ▷ Performance Optimization
- ▷ Process Flow
- ▷ Services
- ▷ Systems
- ▷ Maintenance
- ▷ Web Server
- ▷ Applet Configuration
- ▷ Solution Properties

Update Adeptia Suite Settings - Exercise



Setup > Application Settings

Update System Properties

- Load Management
- Kernel Settings
- Performance Optimization
- Process Flow
  - Recovery

Property Name	abpm.transaction.recovery.enable
Value	yes
Description	ProcessFlow recovery option is valid only if queue processor is disabled

Note :- To activate this property after any change, you need to Restart Server.

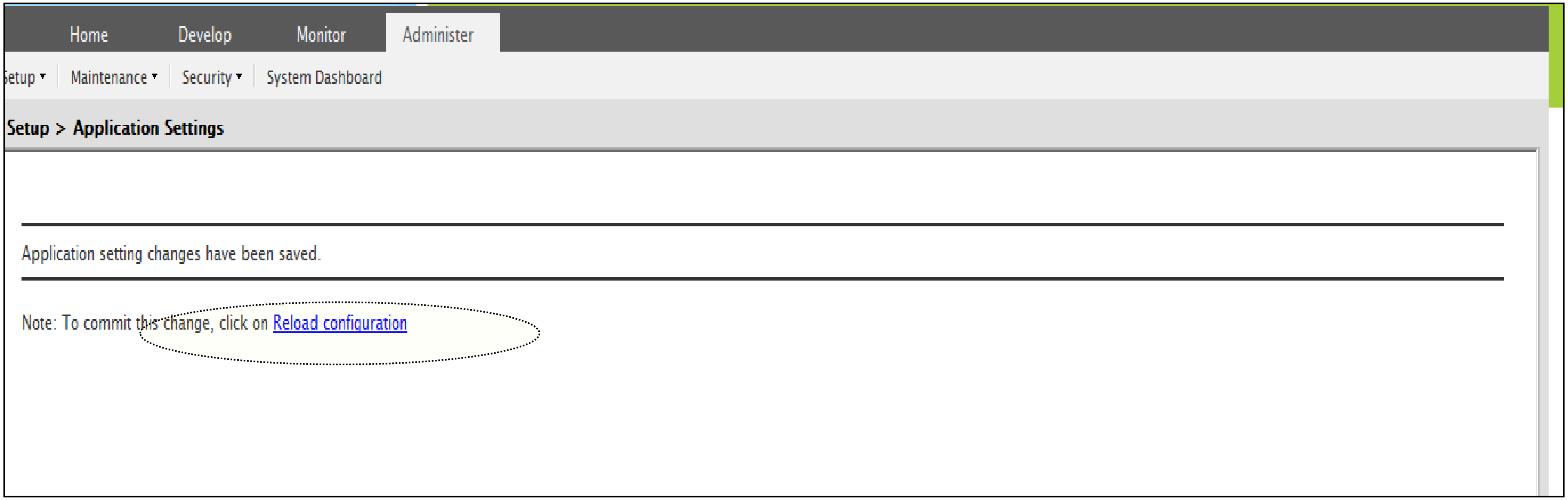
Property Name	abpm.recovery.repository.root
Value	recovery
Description	ProcessFlow recovery path

Note :- To activate this property after any change, you need to Restart Server.

  - Archival
- Services
- Systems
- Maintenance
- Web Server
- Applet Configuration
- Solution Properties

Reload Configuration

Reload Configuration properties – Allows to reload manual configuration property file changes in *AdeptiaServerHome*\ServerKernel\etc\adeptiaserver.properties file.



Setup > Application Settings

Application setting changes have been saved.

Note: To commit this change, click on Reload configuration

Clustering

- Adeptia Suite provides integrated clustering services to deliver higher levels of service and availability.
- Clustering is used for:
  - Load Balancing
  - Fail Over Capability

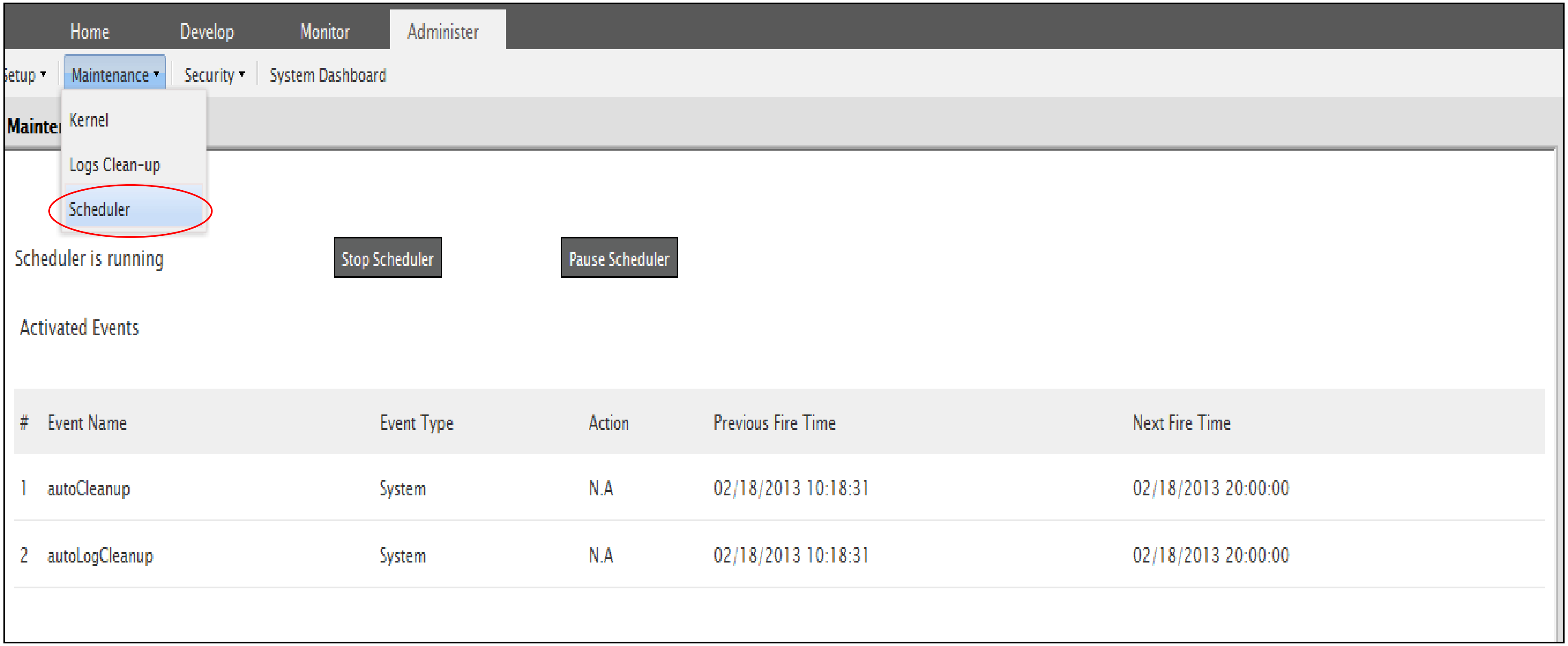
Clustering is enabled by updating the cluster properties in Application settings

Refer to Adeptia Suite User Manual for starting Adeptia Suite in Cluster mode and to view the Cluster status

Start/Stop Scheduler

Allows to start and stop Adeptia scheduler. When the scheduler is turned off, no scheduler related events are supported.

Events like File Event, FTP Event, Mail Event, Calendar Event etc. can be fired.

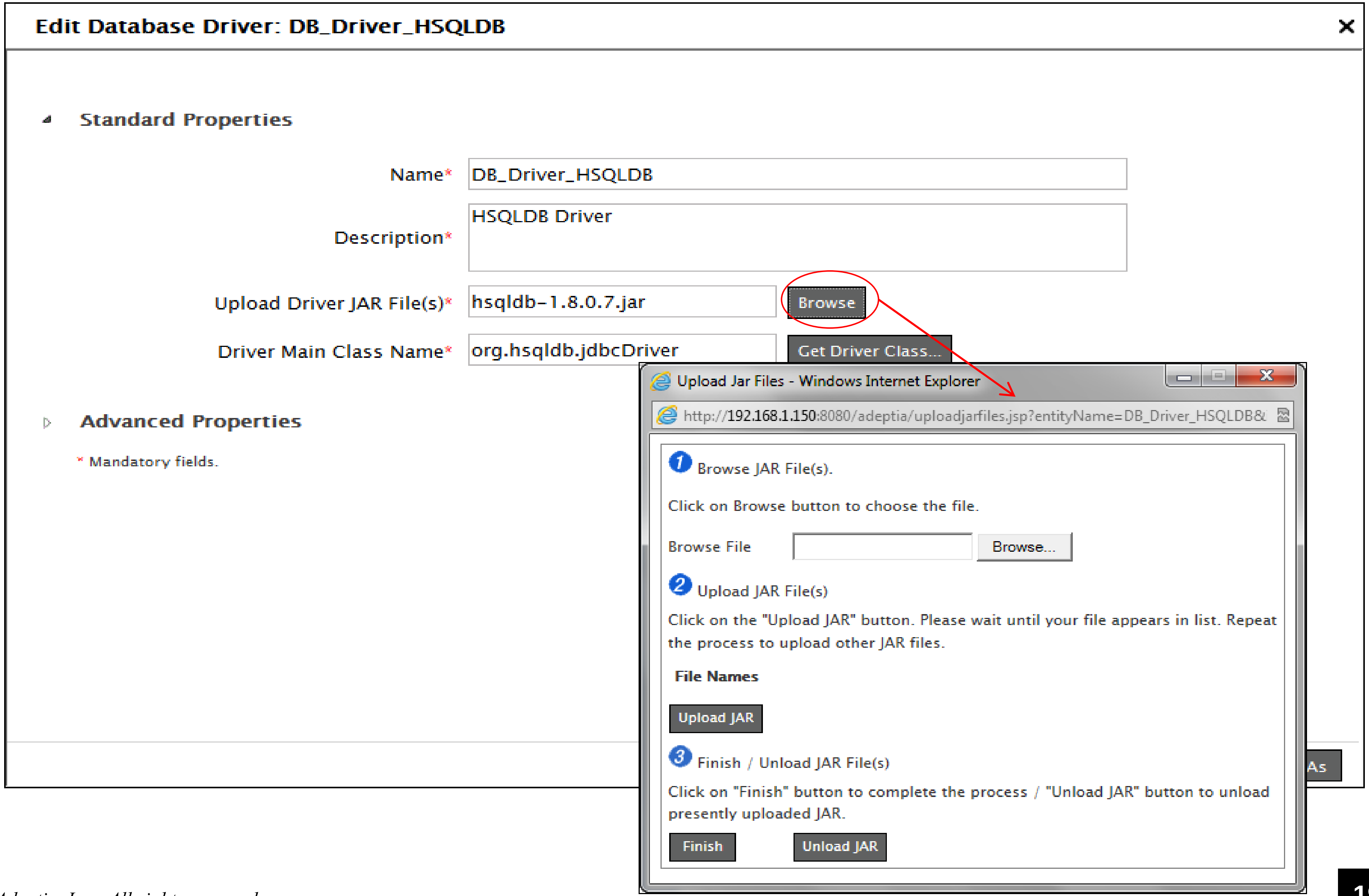


Scheduler is running

Activated Events

#	Event Name	Event Type	Action	Previous Fire Time	Next Fire Time
1	autoCleanup	System	N.A	02/18/2013 10:18:31	02/18/2013 20:00:00
2	autoLogCleanup	System	N.A	02/18/2013 10:18:31	02/18/2013 20:00:00

Database Driver



Edit Database Driver: DB_Driver_HSQLDB

Standard Properties

Name* DB_Driver_HSQLDB

Description* HSQLDB Driver

Upload Driver JAR File(s)* hsqldb-1.8.0.7.jar

Driver Main Class Name* org.hsqldb.jdbcDriver

Advanced Properties

* Mandatory fields.

Upload Jar Files - Windows Internet Explorer

http://192.168.1.150:8080/adeptia/uploadjarfiles.jsp?entityName=DB_Driver_HSQLDB&

1 Browse JAR File(s).

Click on Browse button to choose the file.

Browse File

2 Upload JAR File(s)

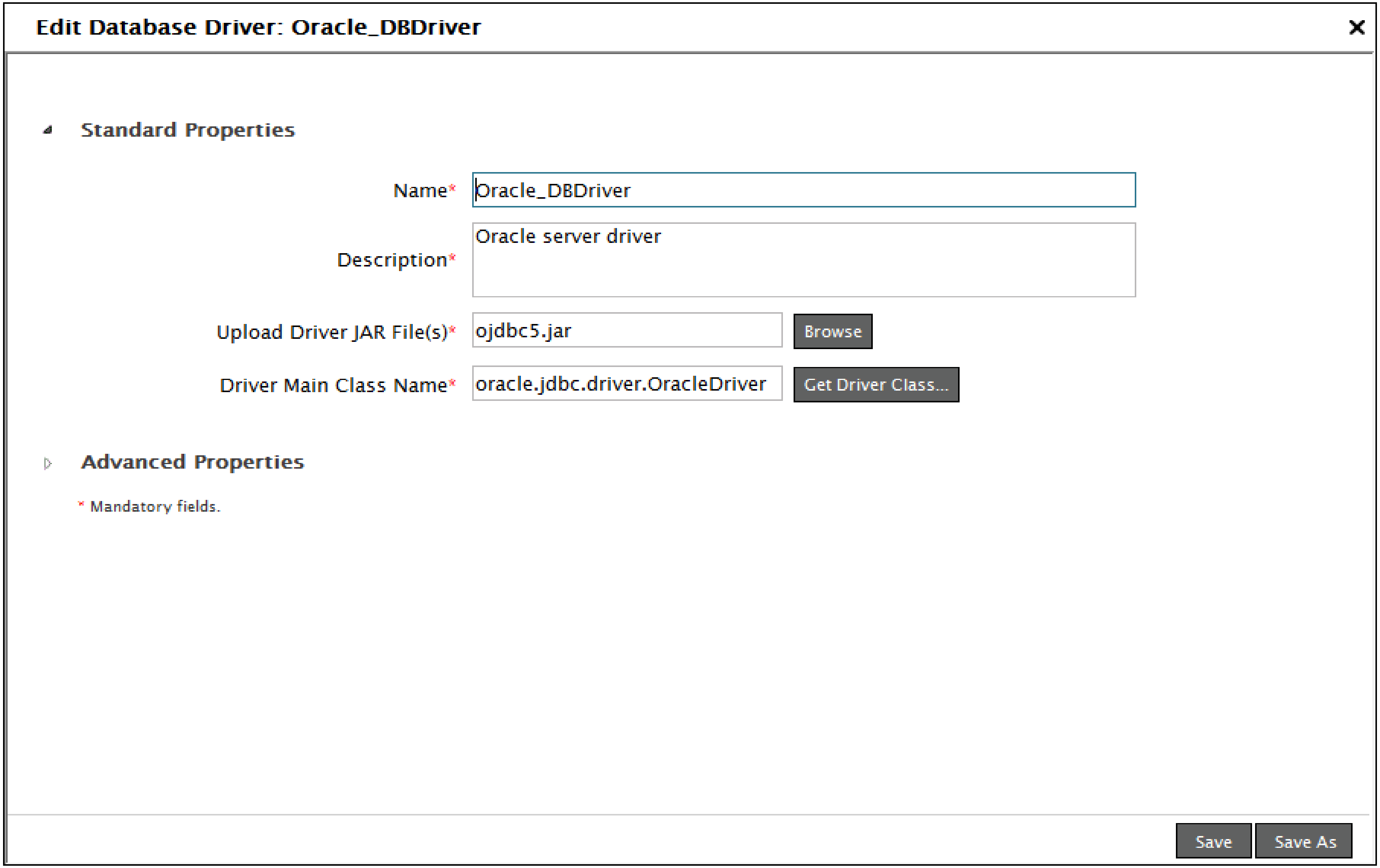
Click on the "Upload JAR" button. Please wait until your file appears in list. Repeat the process to upload other JAR files.

File Names

3 Finish / Unload JAR File(s)

Click on "Finish" button to complete the process / "Unload JAR" button to unload presently uploaded JAR.

Database Driver - Exercise



Edit Database Driver: Oracle_DBDriver

Standard Properties

Name*: Oracle_DBDriver

Description*: Oracle server driver

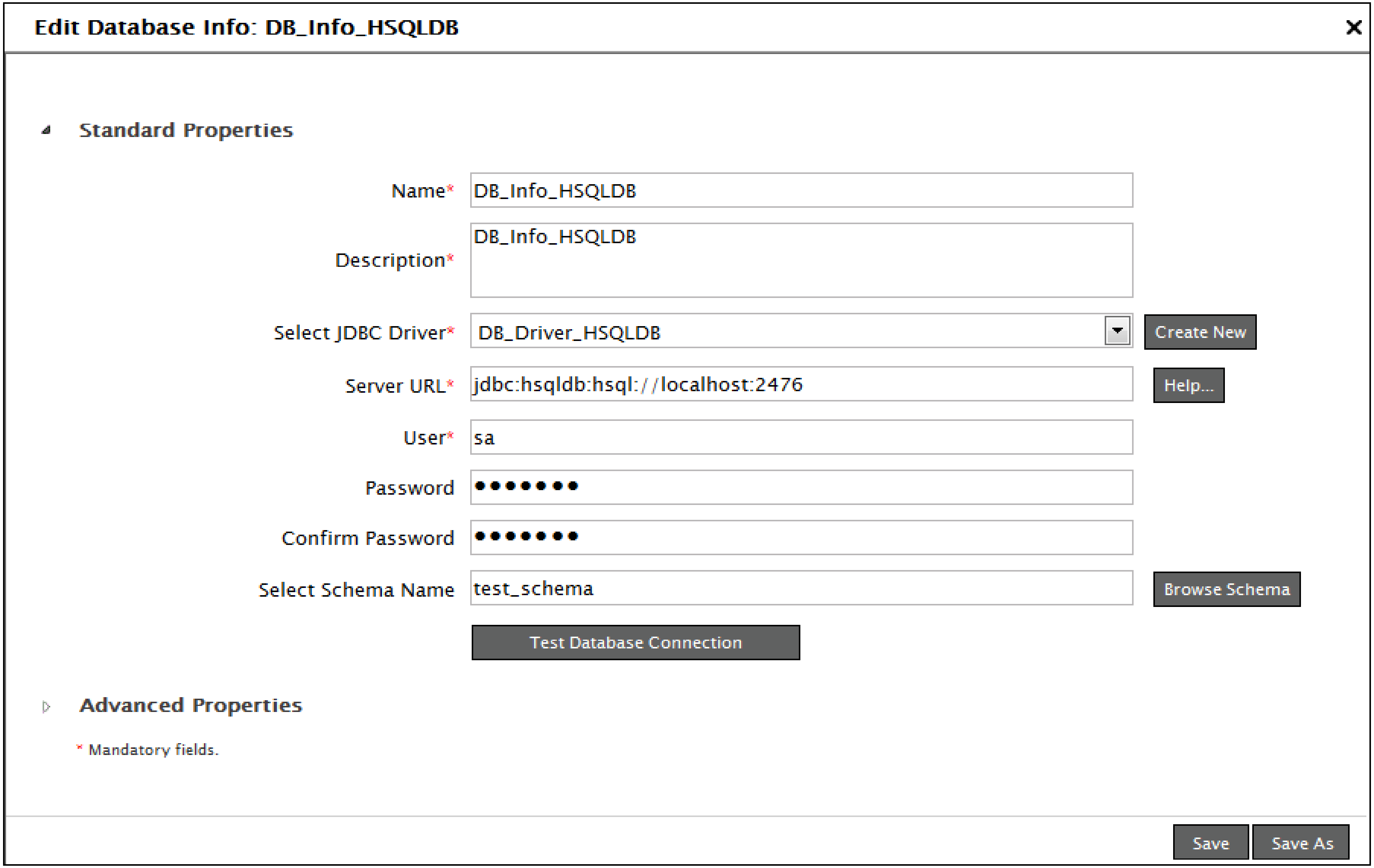
Upload Driver JAR File(s)*: ojdbc5.jar

Driver Main Class Name*: oracle.jdbc.driver.OracleDriver

Advanced Properties

* Mandatory fields.

Database Info



Edit Database Info: DB_Info_HSQLDB

Standard Properties

Name* DB_Info_HSQLDB

Description* DB_Info_HSQLDB

Select JDBC Driver* DB_Driver_HSQLDB

Server URL* jdbc:hsqldb:hsql://localhost:2476

User* sa

Password ••••••••

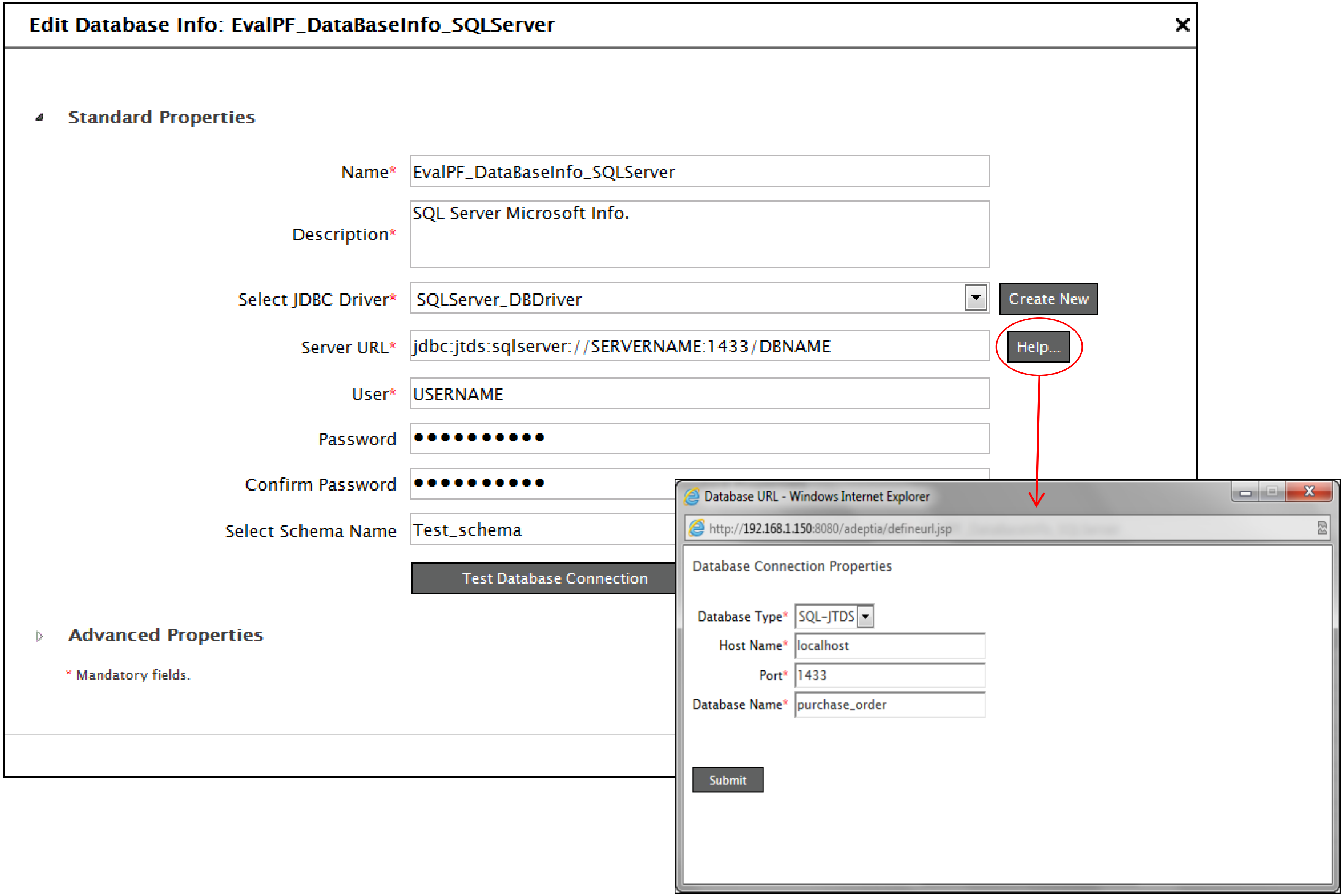
Confirm Password ••••••••

Select Schema Name test_schema

Advanced Properties

* Mandatory fields.

Database Info - Exercise



Edit Database Info: EvalPF_DataBaseInfo_SQLServer

Standard Properties

Name*	EvalPF_DataBaseInfo_SQLServer
Description*	SQL Server Microsoft Info.
Select JDBC Driver*	SQLServer_DBDriver
Server URL*	jdbc:jtds:sqlserver://SERVERNAME:1433/DBNAME
User*	USERNAME
Password	●●●●●●●●●●●
Confirm Password	●●●●●●●●●●●
Select Schema Name	Test_schema

* Mandatory fields.

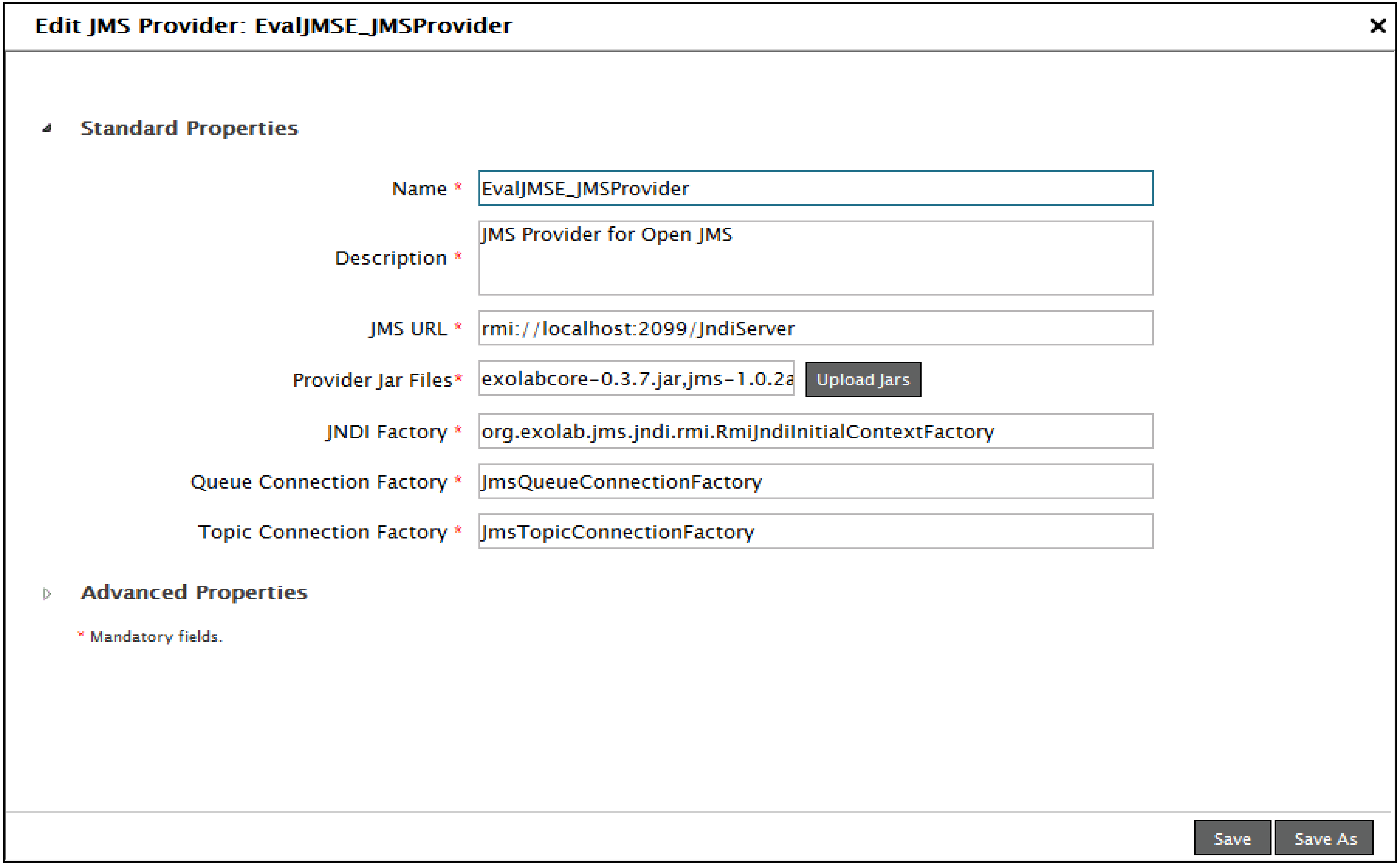
Database URL - Windows Internet Explorer

http://192.168.1.150:8080/adeptia/defineurl.jsp

Database Connection Properties

Database Type*	SQL-JTDS
Host Name*	localhost
Port*	1433
Database Name*	purchase_order

JMS Provider



Edit JMS Provider: EvalJMSE_JMSProvider

◢ Standard Properties

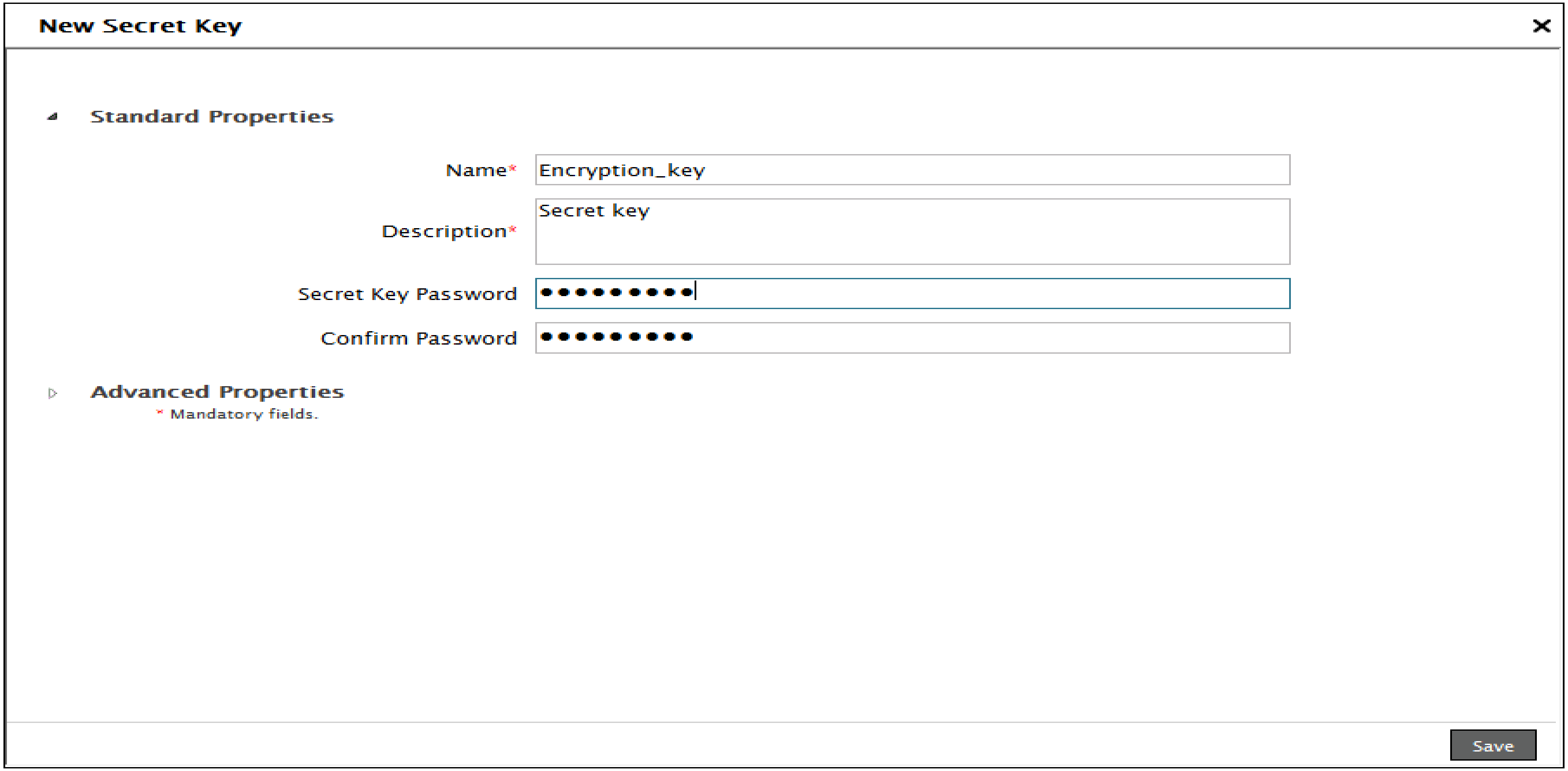
Name *	EvalJMSE_JMSProvider
Description *	JMS Provider for Open JMS
JMS URL *	rmi://localhost:2099/JndiServer
Provider Jar Files*	exolabcore-0.3.7.jar,jms-1.0.2a
JNDI Factory *	org.exolab.jms.jndi.rmi.RmiJndiInitialContextFactory
Queue Connection Factory *	JmsQueueConnectionFactory
Topic Connection Factory *	JmsTopicConnectionFactory

▷ Advanced Properties

* Mandatory fields.

Security – Secret Key

- Secret key is used to create Encryption or Decryption activity.
- Encryption activity is used for encrypting outgoing data. Decryption activity is used for decrypting incoming encrypted data.



New Secret Key

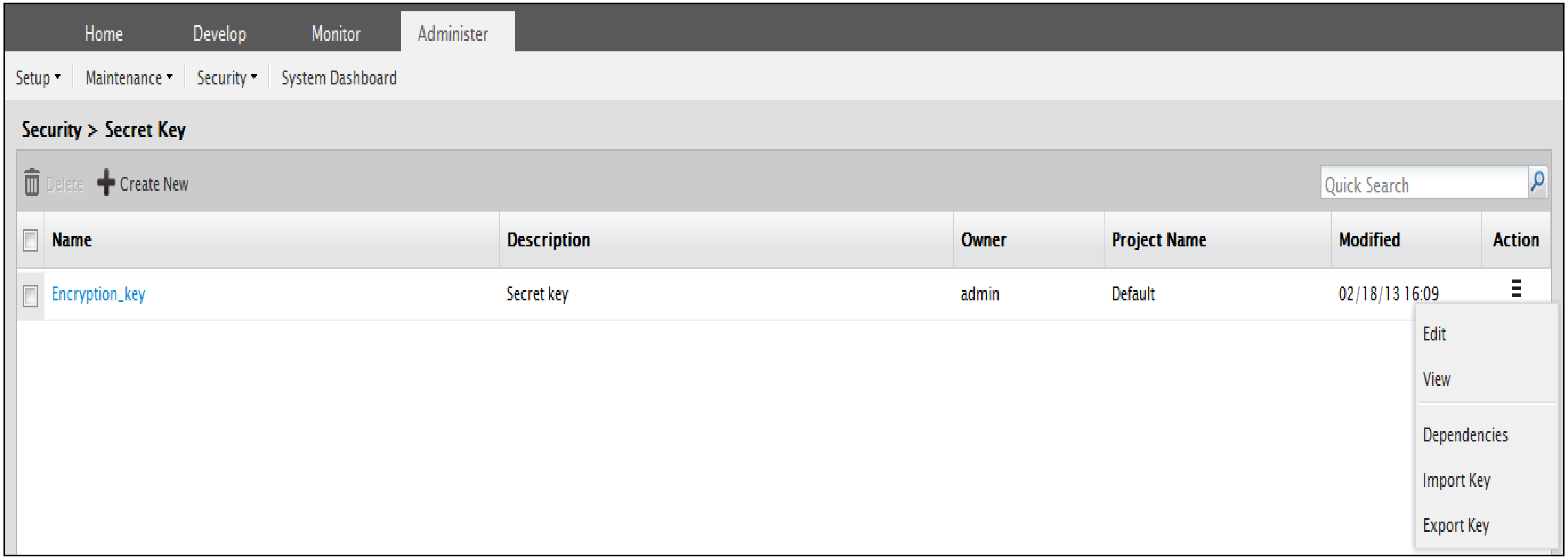
Standard Properties

Name*	Encryption_key
Description*	Secret key
Secret Key Password	••••••••••
Confirm Password	••••••••••

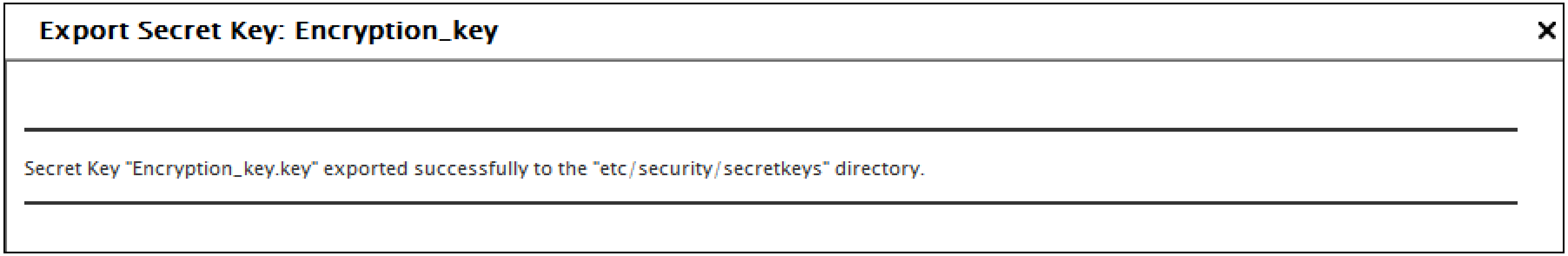
Advanced Properties

* Mandatory fields.

Security – Exporting Secret Key



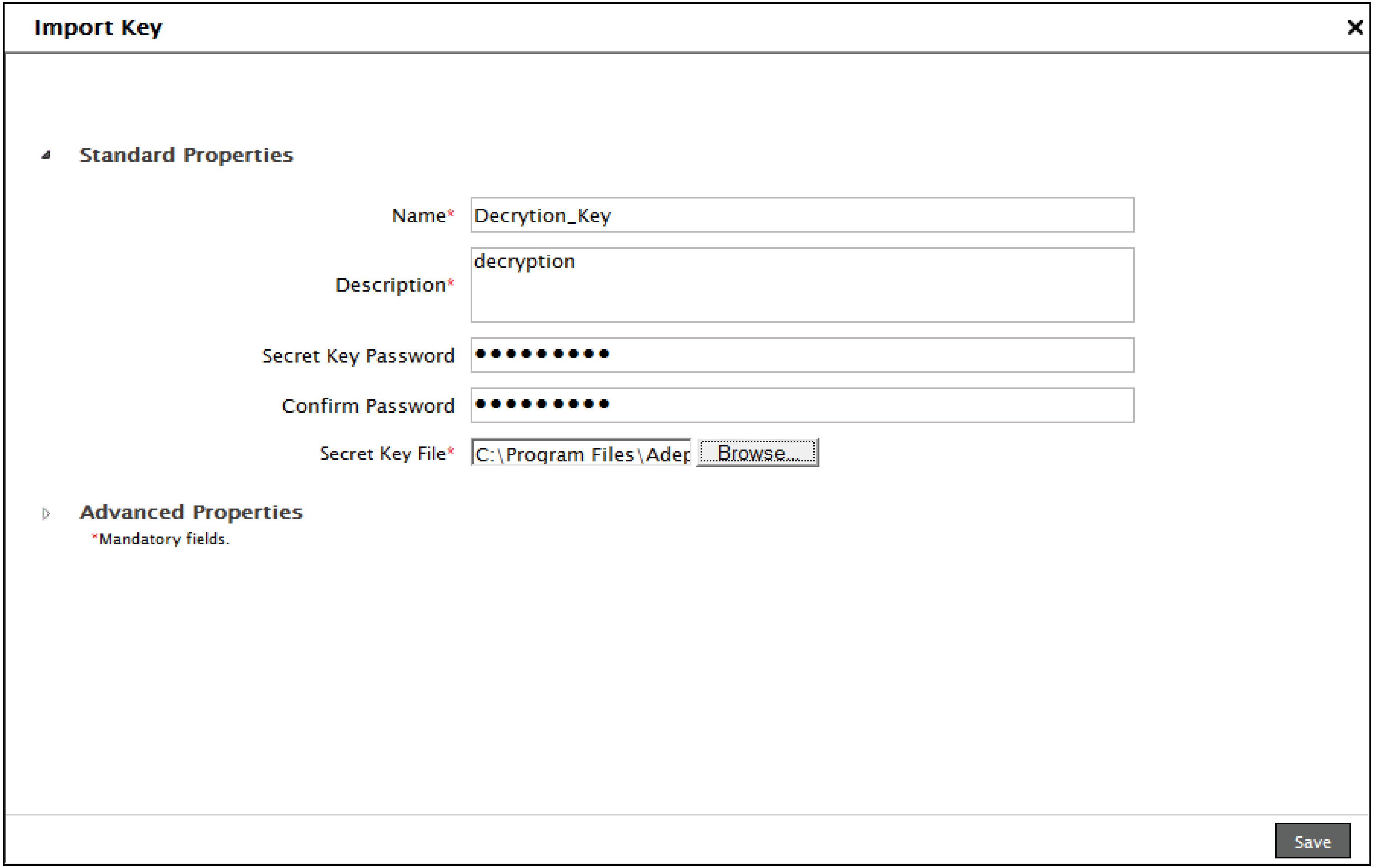
□	Name	Description	Owner	Project Name	Modified	Action
□	Encryption_key	Secret key	admin	Default	02/18/13 16:09	☰



Export Secret Key: Encryption_key ✕

Secret Key "Encryption_key.key" exported successfully to the "etc/security/secretkeys" directory.

Security – Importing Secret Key



Import Key

◢ Standard Properties

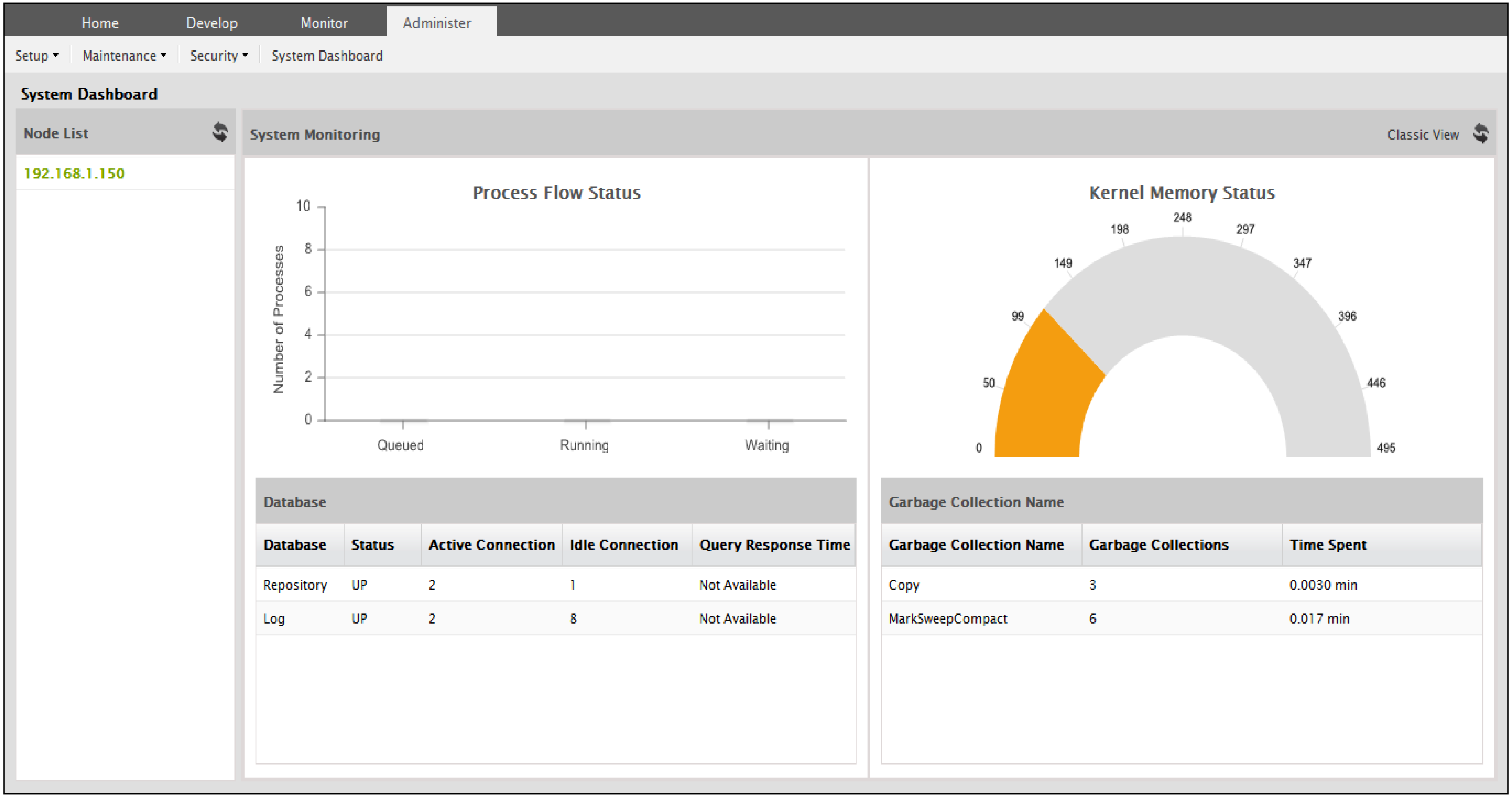
Name*	Decrytion_Key
Description*	decryption
Secret Key Password	••••••••••
Confirm Password	••••••••••
Secret Key File*	C:\Program Files\Adep

*Mandatory fields.

System Status – System Monitor

Allows to view

- Running process flows and their details
- Current User logged in into the Adeptia Suite
- Kernel Memory Usage



Database

Database	Status	Active Connection	Idle Connection	Query Response Time
Repository	UP	2	1	Not Available
Log	UP	2	8	Not Available

Garbage Collection Name

Garbage Collection Name	Garbage Collections	Time Spent
Copy	3	0.0030 min
MarkSweepCompact	6	0.017 min

Q&A

Notes

Notes

For more information contact:

Adeptia Inc.

Web:	www.adeptia.com
Forum:	http://www.adeptia.com/cgi-local/forum/Blah.pl
Email:	support@adeptia.com