5 software AG

4

Create a User Task Activity

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Objectives

- At the end of this section, you will be able to
    - Add a User Task Activity to a process model
    - Generate a simple User Task implementation
    - Deploy the User Task project
    - Interact with the generated User Task interface

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

© Software AG Training   4 - 2

User Task-Oriented BPM

Assignee

Task Execution

Process Execution

Receive Order Doc — Validate Order — Approve Order — Map To Canonical — Insert OPMS — Terminate Process

- A key value of BPM is supporting the interaction between automated process steps and steps requiring human interaction performed by an assignee
- Processes interact with human tasks through User or Manual Task Activities

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

User Task Assignment

- User Tasks may be assigned to humans as part of a process model

- Examples of User Tasks assigned from a process:
  - Expense report approval
  - Check a purchase order
  - Trouble ticket assignment
  - Customer service responsibilities
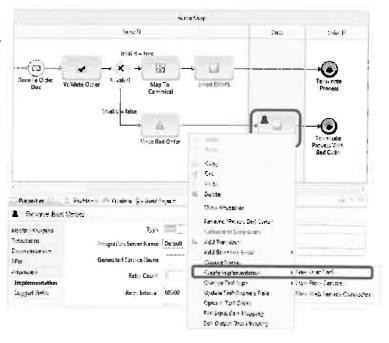 - Resolve issues with a resume format

Software AG Training | 4 - 4

Notes: _____

_____

_____

_____

_____

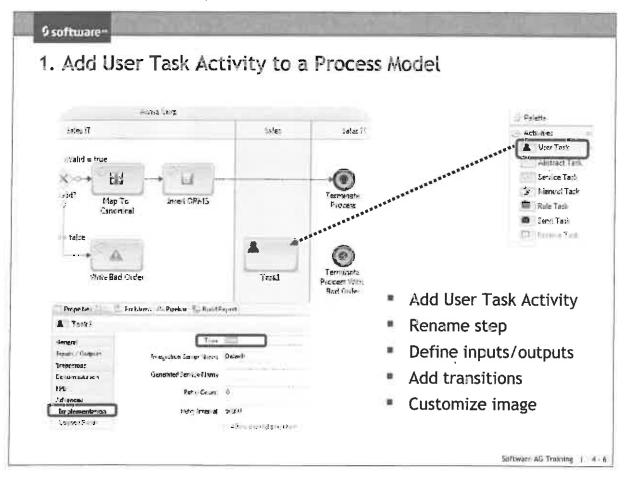_____

_____

_____

_____

_____

_____

_____

Creating a User Task from Scratch - Overview

1. Add an User Task Activity to your process model and assign Inputs/Outputs to your User Task Activity
2. Generate a User Task implementation in User Task Project
3. Customize the User Task Data view
4. Save and Publish User Task project to MWS
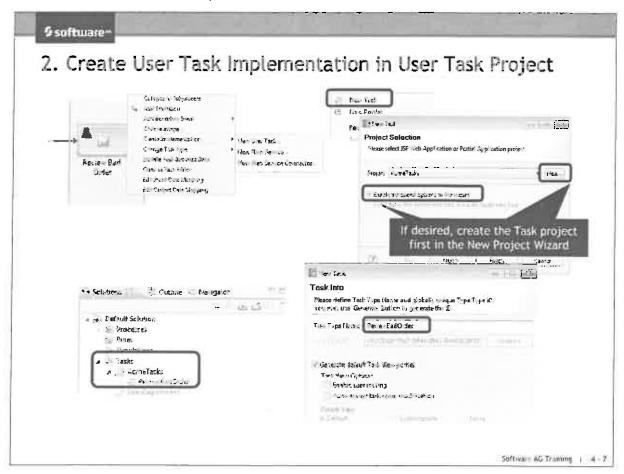5. Build and upload the related process project
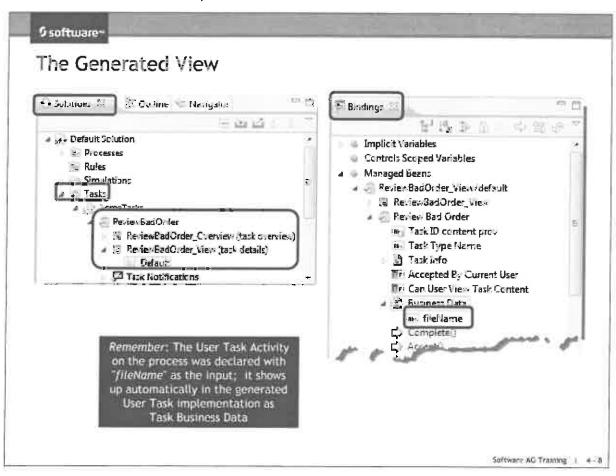6. Test your User Task

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

ϟ software

1. Add User Task Activity to a Process Model

- Add User Task Activity
- Rename step
- Define inputs/outputs
- Add transitions
- Customize image

Software AG Training | 4 - 6

Notes:

2. Create User Task Implementation in User Task Project

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

© Software AG Training   4 - 7

The Generated View

Remember: The User Task Activity on the process was declared with "fileName" as the input; it shows up automatically in the generated User Task implementation as Task Business Data

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

3. Customize the User Task Data View

a) Ensure the view file is opened in the Design canvas (Double click the file highlighted in the Solutions view on the previous page)

b) Locate the business data in the Bindings view

c) Drag the desired business data onto the view

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

4. Publish the User Task Project to MWS

Add the project to the list of projects associated with a running server in the Servers view

The application is deployed to the MWS Server

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

6. Test your User Task...

Check the workspace preferences and ensure the debugger provides you with sufficient time to perform the User Task Activities

A browser shows the User Task in the queue

The business data from the process is displayed

Accept the User Task

Notes:

...6. Test your User Task

After the User Task is accepted, buttons are provided for complete, etc.

Click "Complete" to force the Task Engine to give control back to the Process Engine

In Designer Trace view, the process should continue to the next step in the process after the User Task is completed

User Task shows up as completed

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

4 - 12

5 software AG

5

Process Administration and Monitoring

Notes: _____

_____

_____

_____

_____

_____

_____
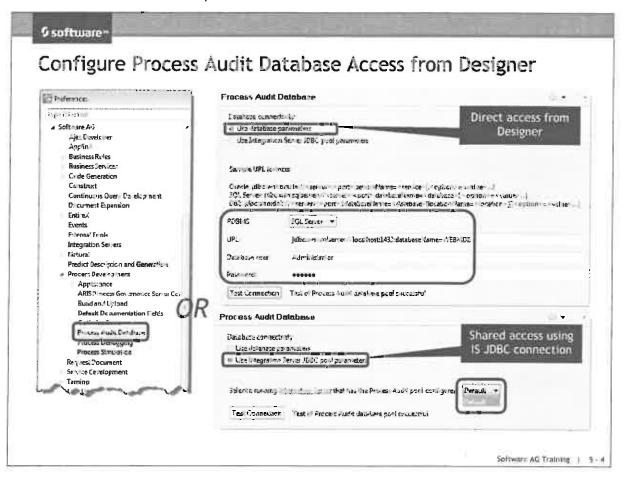
_____

_____

_____

_____

_____

Objectives

- At the end of this section, you will be able to
  - Describe the usage of the Process Audit Database
  - Administrate process models using My webMethods
  - Understand the webMethods Monitor architecture
  - Monitor process instances using My webMethods
  - Resubmit process instances

Software AG Training | 5 - 2

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Processes in the Process Audit Database

- Process Audit database stores information about process models and process instances:
  - When a process is built, information about the process is uploaded into the Process Audit Database by Designer
  - When a process is executed, information about the process instance is written into the Process Audit Database by IS
  - Designer and Integration Server must have access to the database

- Process Audit database stored in DB tables which must be preconfigured. Create appropriate tables using:
  - Database Component Configurator (application or command line mode)
  - delivered DB scripts under:
    ...\SoftwareAG\common\db\scripts

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Configure Process Audit Database Access from Designer

Software AG Training | 5 - 4

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

© Software AG Training   5 - 4
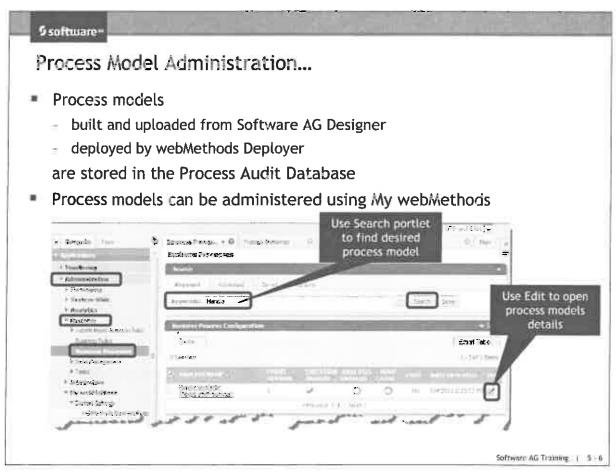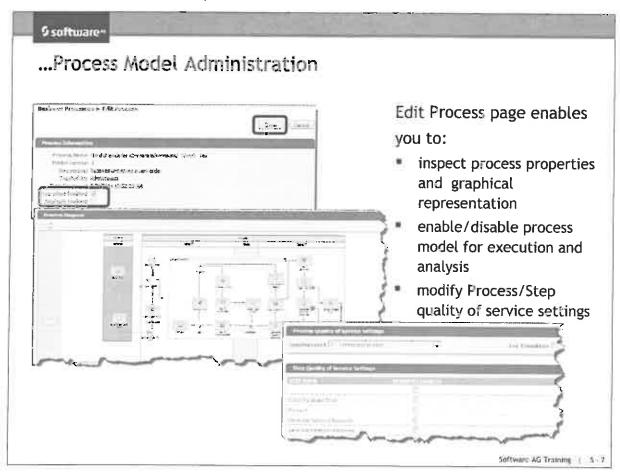
Configure Process Audit Database Access from IS

- IS Access to Process Audit DB configured by IS Administration Console
- Create and assign JDBC Pool to Process Audit component

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Process Model Administration...

- Process models
    - built and uploaded from Software AG Designer
    - deployed by webMethods Deployer

    are stored in the Process Audit Database
- Process models can be administered using My webMethods

Use Search portlet to find desired process model

Use Edit to open process models details

Software AG Training | 5 - 6

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

...Process Model Administration

Edit Process page enables you to:

- inspect process properties and graphical representation
- enable/disable process model for execution and analysis
- modify Process/Step quality of service settings

Software AG Training | 5 - 7

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Monitoring Performed by webMethods Monitor

- webMethods components log data about
  - Business processes
  - Services
  - Documents
  - Custom assets

  to preconfigured database tables

Integration Server

Audit Subsystem

WmMonitor

My webMethods Server

User Interfaces for Monitor & Optimize

Process Audit

IS Core Audit

Process Tracker

Optimize (log data for External Processes)

Configure tables using the *Database Component Configurator*.
The database tables have been previously setup in your class environment.

Software AG Training | 5 - 8

Notes:

webMethods Monitor Capabilities

- If business processes (and services) are set up to log the pipeline data, an administrator can:
    - optionally alter the process (or service) pipeline
    - resubmit business processes (and those services)

- Using the document monitoring capabilities an administrator can:
    - view logged document data
    - resubmit documents
    - optionally edit a document before resubmitting it

> We shall be concentrating on developing processes in such a way that they log pipeline data and also how to test processes if data is resubmitted.

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

⑤ software™

Monitoring Processes

webMethods Monitor provides the ability to monitor business processes:

> This course has been emphasizing webMethods-executed processes

- **webMethods-executed processes** are modeled using Designer and execute in the webMethods platform

- **Externally executed processes** are modeled using Designer as a Business Analyst but they execute outside of the webMethods platform
  - For example, an externally executed process may execute process steps in Tomcat, BEA WebLogic or a mainframe
  - The steps executing in external systems must have coded "hooks" to communicate process state to MWS Monitor

- **Integration processes** execute as a series of Integration Server services that directly invoke each other.
  - No Designer model exists for an Integration process
  - Hooks must be added to the Integration Server process to log the process state to the audit log
  - Leverage the `pub.monitor.integrationProcessLogging` service from the WmMonitor package

Software AG Training | 5 - 10

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Process Logging Level

- Process logging dependent on the logging level
  - Process logging level is preset as part of the process properties
  - Set using Designer

- Process logging level determines:
  - Amount of information the process engine writes
  - Actions available from My webMethods

- Process logging level can be changed by MWS Administrators using My webMethods Administration

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____
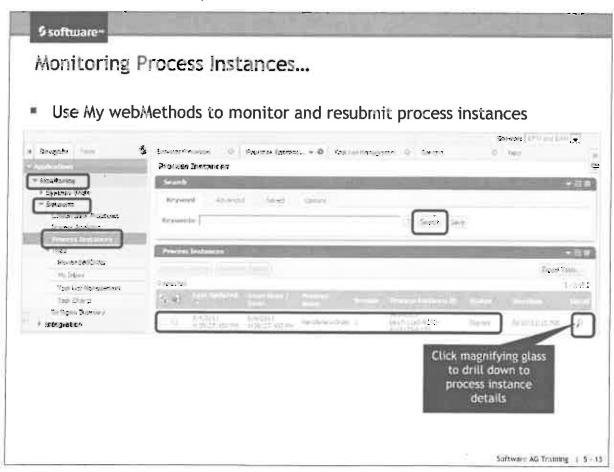
Process Logging Levels and Actions
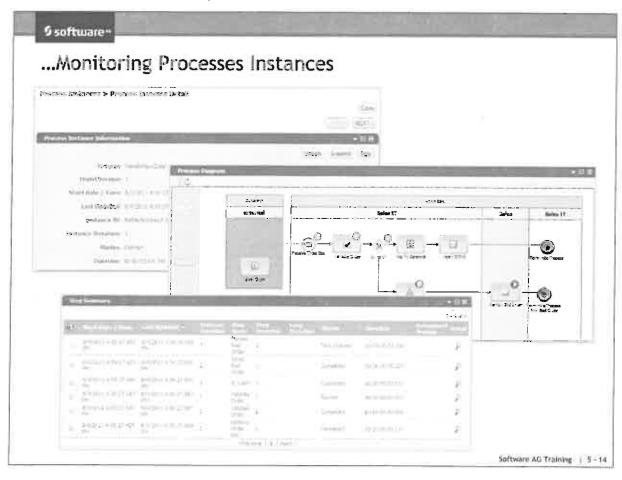
Logging Level	Description	View Process Status	View Step Status	Resubmit Process
1 - None	Disables logging	No	No	No
2 - Errors only	Logs process status when steps fail. Logs input pipeline for failed steps	At failed step	No	At failed step
3 - Process only	Logs process status. Logs input pipeline for failed steps	Yes	No	At failed step
4 - Process and Start Events	Logs process and Start Event status. Logs input pipeline for Start Events and failed steps	Yes	For Start Events	At Start Event or failed step
5 - Process and all Events and Activities	Log process and all steps status. Logs input pipeline for all steps excluding looping activities	Yes	For all steps	At any step that logged input pipeline
6 - Process and all Events, Activities and looped Activities	Log process and all steps status. Logs input pipeline for all steps including looping activities.	Yes	For all steps	At any step that logged input pipeline

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Monitoring Process Instances...

- Use My webMethods to monitor and resubmit process instances

Click magnifying glass
to drill down to
process instance
details

Software AG Training | 5 - 13

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

5 - 13

...Monitoring Processes Instances

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

5 - 14

Step Status and Actions

- webMethods Monitor displays the status of each step in the process instance:
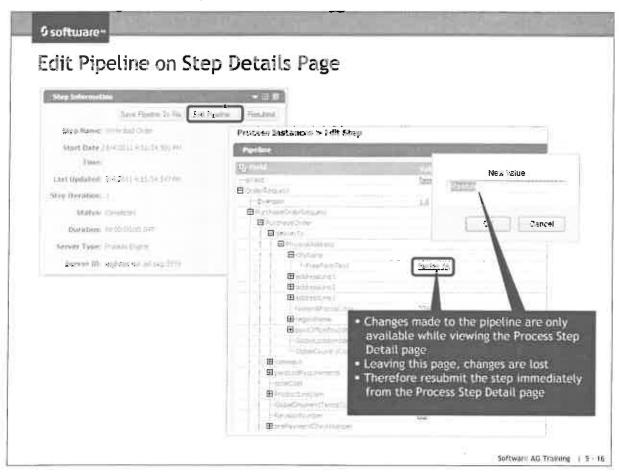 - The step is currently running
  - The step has successfully completed
  - The step failed
  - The step is waiting - for example, waiting for a Broker document

- From the step detail page:
  - Pipeline for that step may be saved to a file
  - Pipeline may be edited before resubmitting the process or step
  - Possibility to restart the process step

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Edit Pipeline on Step Details Page

- Changes made to the pipeline are only available while viewing the Process Step Detail page
- Leaving this page, changes are lost
- Therefore resubmit the step immediately from the Process Step Detail page

Notes:

Process Resubmission

- Processes can only be resubmitted if the process is in status:
  - Completed
  - Failed
  - Resumed
  - Stopped (Canceled)

- Use My webMethods to resubmit from a process step:
  - Applications -> Monitoring -> Business -> Process Instances -> Process Instance Detail
    -> Step Summary -> Process Step Detail
  - Allows a process to be resubmitted "from the middle"
  - Process pipeline can be edited before resubmission

- External and integration processes cannot be resubmitted

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

9 software

Resubmission Considerations

- A process with a queued User Task should be stopped prior to resubmission of a step upstream from the User Task Activity, causing the current User Task instance to be automatically cancelled

- The Process Engine starts a new iteration of the process with the same process ID and executes from the resubmitted step onwards

- New instances of any subsequent User Tasks will be created when reached

- Monitoring the process instance in its new iteration will not show logs for steps that were upstream from the resubmitted step

Software AG Training | 5 - 18

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

♦ software·

6

Parallelism and Joins

Notes: ————————————————————————

—————————————————————————————————

—————————————————————————————————

—————————————————————————————————

—————————————————————————————————

—————————————————————————————————

—————————————————————————————————

—————————————————————————————————

—————————————————————————————————

—————————————————————————————————

—————————————————————————————————

—————————————————————————————————

—————————————————————————————————

Objectives

- At the end of this section, you will be able to
    - Describe the different ways of parallelism that can be implemented
    - Implement parallelism using parallel steps and Gateways
    - Distinguish between the different Join types
    - Hide process complexity and parallelism using Subprocesses

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Parallelism

- Business processes are typically long lived and often have associated service level agreements (SLAs)

- All steps in a process are not necessarily dependent on all up-stream process steps
  - Profile processes and leverage parallelism to increase performance and throughput.

- Parallelism may be designed in several ways:
  1. Asynchronous Web services
  2. Threaded remote service calls
  3. Publish/subscribe using Broker or JMS
  } **Not covered in this BPM class**
  4. Parallel steps, Joins and Subprocesses in a distributed process runtime

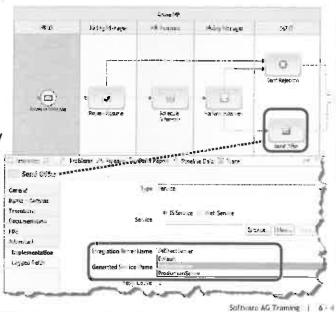- Designing with parallelism is key to meeting tight SLA requirements

Software AG Training | 6 - 3

Notes: _____

_____

_____

_____

_____

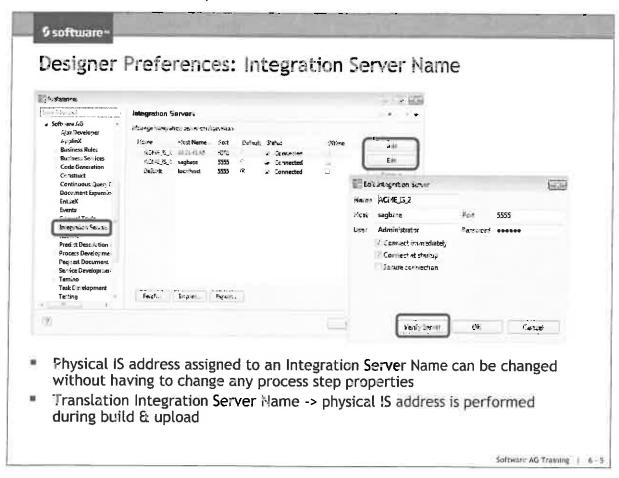_____

_____

_____

_____

_____

_____

_____

_____

♦ software™

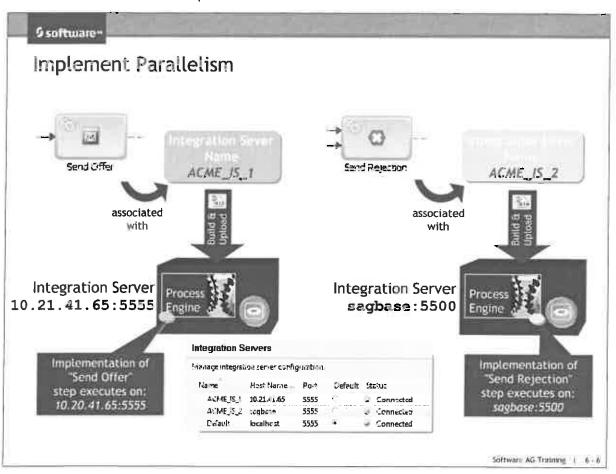## Integration Server Names and Parallelism

- Process model steps must be assigned to an **Integration Server Name.**
- Each Integration Server name must be associated with a **physical integration Server**
- Integration Server names provide an abstraction layer between model steps and physical Integration Servers
- Process steps are executed by **Process Engine** hosted by the associated physical IS
- Enables you to distribute process execution across multiple integration Servers and to implement parallelism

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

webMethods 8.2 BPM for Developers

Designer Preferences: Integration Server Name

- Physical IS address assigned to an Integration Server Name can be changed without having to change any process step properties
- Translation Integration Server Name -> physical IS address is performed during build & upload

Notes:

Implement Parallelism

Send Offer

Integration Server
Name
ACME_IS_1

associated
with

Integration Server
10.21.41.65:5555

Process
Engine

Implementation of
"Send Offer"
step executes on:
10.20.41.65:5555

Send Rejection

Integration Server
Name
ACME_IS_2

associated
with

Integration Server
sagbase:5500

Process
Engine

Implementation of
"Send Rejection"
step executes on:
sagbase:5500

Integration Servers

Manage integration server configuration

Name	Host Name	Port	Default	Status
ACME_IS_1	10.21.41.65	5555		Connected
ACME_IS_2	sagbase	5555		Connected
Default	localhost	5555	•	Connected

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

ⓢ software~

Process Pipeline Properties

- Process properties determine how the process pipeline will be handled at runtime
- To access Process properties, click white space around the process pools and select Properties – Run Time tab

Optimize Locally: Services in the same IS will be executed without leveraging the Broker or internal dispatcher

Express Pipeline: If steps in a process execute in multiple Integration Servers, the transition document will only contain the data required by the downstream steps, not the entire process pipeline

Volatile Transition Documents: The IS does not store transition documents on its file system before forwarding them to the recipient. If the IS crashes before the delivery, the transition documents are lost

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Optimize Locally - The Expected Behavior

IF   Consecutive steps in a process are:
- sequential
- not User Task- related
- executed on the same Integration Server

☑ Optimize Locally

AND   "Optimize Locally" is enabled

THEN   Process Engine uses a single thread to:
- execute the process steps
- perform logging
- avoid using the Broker and internal dispatcher

Software AG Training | 6 - 9

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

$ software™

Optimize Locally - The Unexpected Behavior

☑ Optimize Locally

- If steps in a process are modeled as <u>parallel</u>, messages will pass through the Broker and dispatcher even if they are implemented on one physical Integration Server
  - `ProcessTransitionVol` messages are published regardless of the setting of "Optimize Locally"
  - The generated trigger configured to fire on receipt of a `ProcessTransitionVol` document is set to:
    - Processing Mode: Concurrent
    - Max Execution Threads: 10

- Therefore, parallel steps in a process execute in parallel threads even if they exist on the same Integration Server

- Ensures that parallel steps are executed truly parallel

Software AG Training | 6 - 10

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Optimize Locally at Process Runtime - Sample

- Assuming all steps are executed on the same physical IS with "Optimize Locally" set to TRUE:

No documents published

Documents published to allow parallel processing

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Allow Parallel Execution Property

- Enable "Allow parallel execution" property if parallelism is possible
- Available for all Activities*, Events, and Gateways
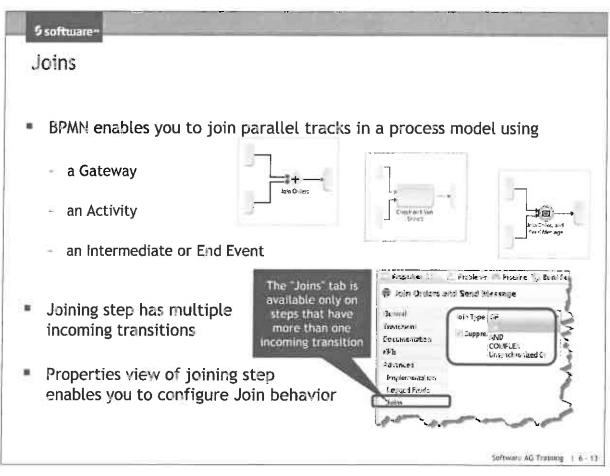
* Exception: Manual Task Activity

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Joins

- BPMN enables you to join parallel tracks in a process model using

  - a Gateway

  - an Activity

  - an Intermediate or End Event

- Joining step has multiple incoming transitions

> The "Joins" tab is available only on steps that have more than one incoming transition

- Properties view of joining step enables you to configure Join behavior

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Joins Types in webMethods BPMS
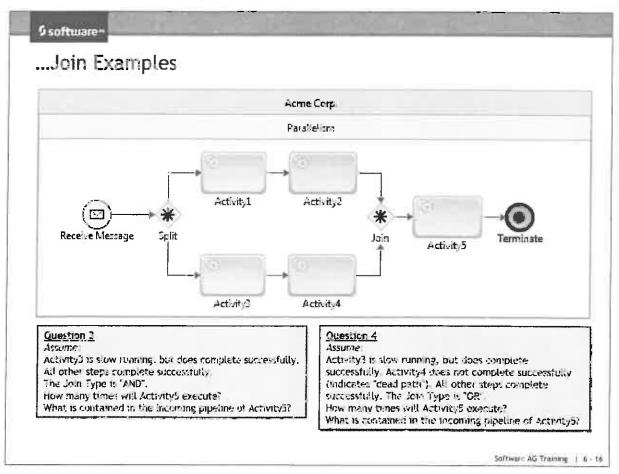
Join Type	Description	Additional Information
AND	All incoming transitions required to make process continue. Pipeline will be merged.	- opt. Join Timeout transition (BPMN extension) - opt. Unsatisfied Join transition (BPMN extension)
OR	At least one incoming transition required to make process continue. Pipeline will be merged.	- opt. Unsatisfied Join transition (BPMN extension)
Unsynchronized OR	Any incoming transition will cause the process to continue. Step may "fire" multiple times. No pipeline merging.	-webMethods proprietary (BPMN extension)
COMPLEX	The join is based on conditional logic.	- Complex Join Condition - opt. Join Timeout transition (BPMN extension) - opt. Unsatisfied Join transition (BPMN extension)

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

6 - 14

Join Examples...

Acme Corp.

Parallelism

Receive Message → Split → Activity1 → Activity2 → Join → Activity5 → Terminate

Activity3 → Activity4

Question 1
Assume:
Activity3 is slow running, but does complete successfully.
All other steps complete successfully.
The Join Type is "OR".
How many times will Activity5 execute?
What is contained in the incoming pipeline of Activity5?

Question 2
Assume:
Activity3 is slow running, but does complete successfully.
All other steps complete successfully.
The Join Type is "unsynchronized OR".
How many times will Activity5 execute?
What is contained in the incoming pipeline of Activity5?

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

webMethods 8.2 BPM for Developers

....Join Examples

Acme Corp.

Parallelism

Receive Message — Split — Activity1 — Activity2 — Join — Activity5 — Terminate

Activity3 — Activity4

Question 2
Assume:
Activity3 is slow running, but does complete successfully.
All other steps complete successfully.
The Join Type is "AND".
How many times will Activity5 execute?
What is contained in the incoming pipeline of Activity5?

Question 4
Assume:
Activity3 is slow running, but does complete successfully. Activity4 does not complete successfully (indicates "dead path"). All other steps complete successfully. The Join Type is "OR".
How many times will Activity5 execute?
What is contained in the incoming pipeline of Activity5?

Software AG Training | 6 - 16

Notes:

$software

Setting the Join Type

- Join type can be customized at joining
  - Activity
  - Intermediate/End Event
  - Complex Gateway

- Join type fixed when using joining
  - Parallel Gateway
  - Exclusive gateway
  - Inclusive Gateway

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Parallel Gateway

- *Diverging*: All outgoing transitions taken unconditionally

- *Joining*: AND

- No Join timeout available

Parallel Gateway does not allow conditions on output transitions. All transitions will be taken.

Notes:

Exclusive Gateway

- *Diverging:* One or none transition taken conditionally

- *Join:* Unsynchr. OR

- No Join timeout available

PurchaseOrder:PurchaseOrder/NumItems >= 100

Receive PurchaseOrder — Diverging Exclusive

Do Either This

Default — Or Do This

Joining Exclusive — Terminate

Recommended, but not mandatory

Diverging Exclusive
- General
- Inputs / Outputs
- Transitions
- Documentation
- KPIs
- Advanced
- Implementation
- Logged Fields

Migration target Name: Default
Gateway Type: Exclusive
- Complex
- Inclusive
- Exclusive
- Parallel

Joining Exclusive
- General
- Inputs / Outputs
- Transitions
- Documentation
- KPIs
- Advanced
- Implementation
- Logged Fields

Icon Type: Unsynchronized Or

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Inclusive Gateway

- *Diverging*: Zero, one or more transitions taken conditionally

- *Joining*: OR

- "Default" transition recomm.

- No Join timeout available

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Complex Gateway

- For complex merging and branching behavior
- Joining Complex Gateway supports all Join types
- Offers Join timeout for AND and COMPLEX Join types

Software AG Training | 6 - 21

Notes:

Join Timeout

- Join timeouts* specify how long a joining step will wait after the first transition reaches it and before it continues down stream
- Only available for joining Activities, Events and Complex Gateways with Join types AND or COMPLEX

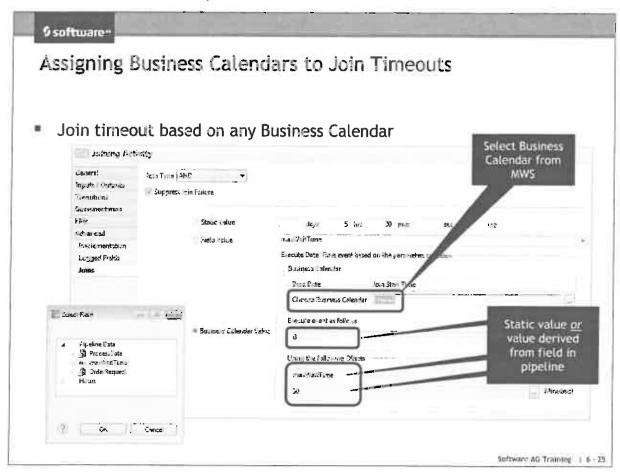* webMethods extension to BPMN specification

Notes:

Join Timeouts based on Business Calendars

- *Option*: Join timeout value based on Business Calendar as opposed to clock timeouts

- *Sample*:

Join Timeout of one business hour

Join started at 10:00PM

Employee start work at 8:00AM

Join expires at 9:00AM

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

brief

Configuration of Business Calendars

- **Create** and maintain reusable Business Calendars from My webMethods Administration
- Multiple Calendars for different shifts and countries

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Assigning Business Calendars to Join Timeouts

- **Join timeout based on any Business Calendar**

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Join Timeout Transitions

- If a step has a configured Join Timeout this enables you to add an outgoing conditional transition of type "Join Timeout"*
- Timeout condition displayed as annotation
- Helps to create robust processes

webMethods extension to BPMN specification

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

6 - 26

$ software™

Join Synchronization (AND, OR)

1. Joining step with Join type *AND* or *OR* is started by the first incoming transition
2. Step waits for live or dead paths of all other incoming legs
   - Live path: incoming leg is used and reaches the joining step
   - Dead path: Process Engine detects at runtime that this leg is not used or terminates with an error
3. After receiving all live/dead paths
   - joining step merges pipelines of all received live paths
   - IF
     - *AND*: all incoming legs arrived with a live path
     - *OR*: at least one leg arrived with a live path
   - THEN
     - Joining step becomes satisfied and fires all non-exceptional outgoing transitions passing the merged pipeline
   - ELSE
     - Join becomes "unsatisfied" and fires Unsatisfied Join transition, if modeled
4. Joining step completes with status "Completed" or "Unsatisfied Join"

Check and Join Orders

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

© Software AG Training   6 - 27

ⓢ software™

Unsatisfied Join Transition

- Joining step with join type AND, OR or COMPLEX enables you to add an outgoing conditional transition of type "Unsatisfied Join"*

- Transition only used if joining step is unsatisfied

- Not used in case of a Join timeout

- Provided description or "Unsatisfied Join" displayed as annotation at the transition

- Helps to create robust processes

* webMethods extension to BPMN specification

Notes:

Unsatisfied Joins and Suppress Join Failure Option

- "Suppress Join Failure" option available at joining step with join type AND, OR or COMPLEX
- Instructs the Process Engine how to handle an Unsatisfied Join:

	Unsatisfied Join transition modeled	Unsatisfied Join transition not modeled
Suppress Join Failure checked (default)	Process terminates at joining step without an error; no outgoing transition used	Process terminates at joining step without an error; no outgoing transition used
Suppress Join Failure unchecked	Process continues without an error and uses Unsatisfied Join transition	Process terminates at joining step with error

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

$ software

Subprocesses

- Subprocess is a BPMN Activity allowing to encapsulate process steps and transitions on the design canvas
- Usage: Structure process models, hide (technical) complexity and parallelism, improve readability
- Can be collapsed and expanded to hide/show contained process logic
    - collapsed subprocess decorated by "+" Activity Marker
    - does not offer a step image
- Label and color can be set in the properties
- May cross multiple swimlanes
- Subprocesses are NOT implemented as separate processes, therefore
    - are therefore NOT reusable
    - must NOT start with a Start Event

Order
Investigation

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____
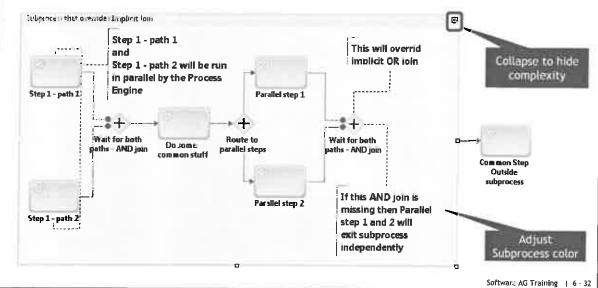
Adding a Subprocess - Two Choices

1. Add empty Subprocess from Palette
   - Add other Activities, Events, Gateways,... to automatically expanded empty Subprocess
   - Model internal, incoming, and outgoing transitions

2. Collapse marked portion of your process model to a new Subprocess

Check Backend   Approve   Investigate Level

Undo
Redo Collapse to Subprocess
Copy
Cut
Paste
Delete
Show Properties
Rename 'Check Backend'
Collapse to Subprocess
Add Boundary Event
Choose Image...
Create Implementation
Change Task Type
Edit Data Mapping

Click to expand

Subprocess1

Rename

Notes:

Subprocesses and Parallelism

- Subprocesses having multiple steps with no **incoming** transitions will run as multiple threads in parallel.
- All steps with no **outgoing** transitions are implicitly joined with an OR
  - Add trailing Gateway or Activity step to override the Join behavior

Notes: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Subprocess with Loop Marker

- Subprocess may have a Loop Marker
- Loop Marker controls multiple execution of the Subprocess
- Available for collapsed and expanded Subprocesses
- Loop configuration using Loop Properties.
- Loops are supported by Debugger and Process Monitor as well.

See next chapter for more details

Software AG Training | 6 - 33

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

5 software™

This page intentionally left blank.

Notes:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____