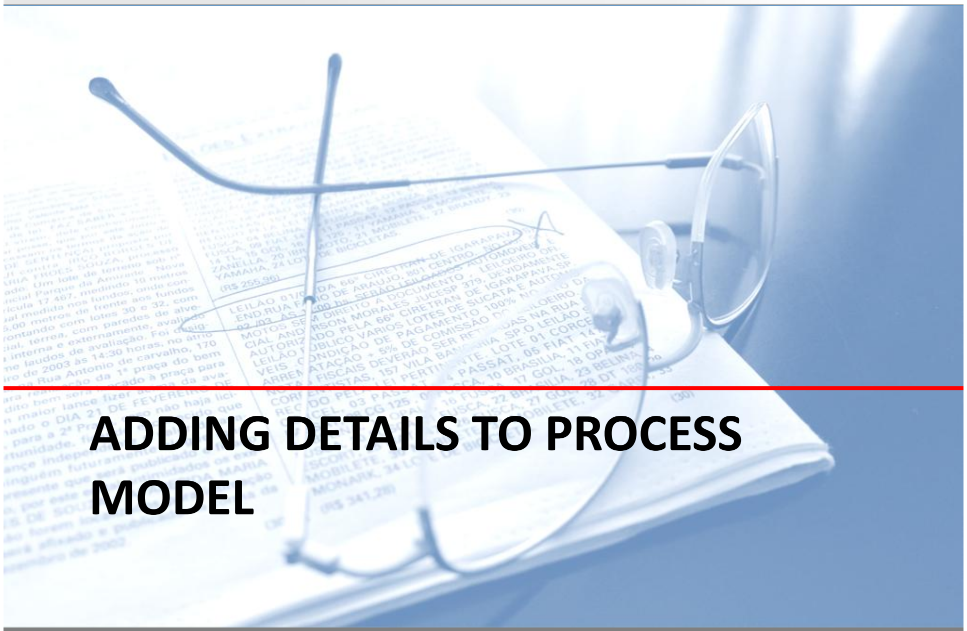


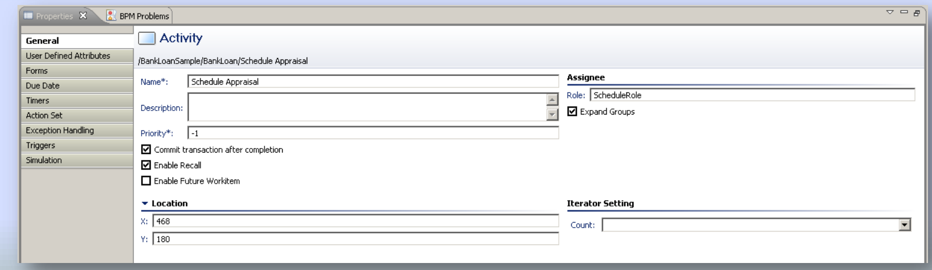
ADDING DETAILS TO PROCESS MODEL

Node Properties

- Node properties
  - General
  - User Defined Attributes
  - Forms
  - Due Date
  - Timers
  - Action Set
  - Exception Handling
  - Triggers
  - Simulation

General Attributes

- Name and Description
- Role
- Transaction Setting
- Expand Group
- Enable Recall
- Future Workitems
- Iterator Setting



Activity

/BankLoanSample/BankLoan/Schedule Appraisal

Name*: Schedule Appraisal

Description:

Priority*: -1

☑ Commit transaction after completion

☑ Enable Recall

☐ Enable Future Workitem

Location

X: 468

Y: 180

Assignee

Role: ScheduleRole

☑ Expand Groups

Iterator Setting

Count:

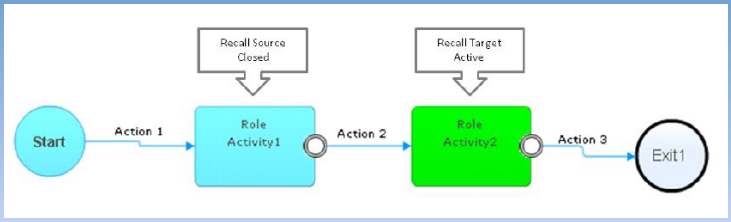
General Attributes

- Transaction Setting
  - Provides the option to commit or hold when task completes
  - Commit may be delayed if data loss is acceptable in case of crashes
- Expand Group
  - Individual work items for each member (Expand Group)
  - Users in the same role share copy of one work item

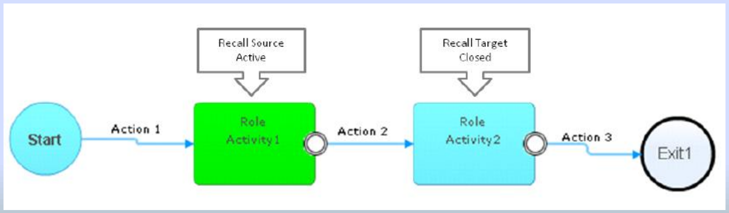
Recalling Task

- Recalling Task allows a user to recall a Task they previously completed, to edit and resubmit, if required.
  - Example: User completes Activity1 and the Process moves to Activity 2. The user may decide to recall the Activity1 task.
- Can only Recall one task at a time
- Can recall multiple levels through iterating each level
  - Example: Recall Activity3, then Activity2 and finally Activity1

Recalling Task

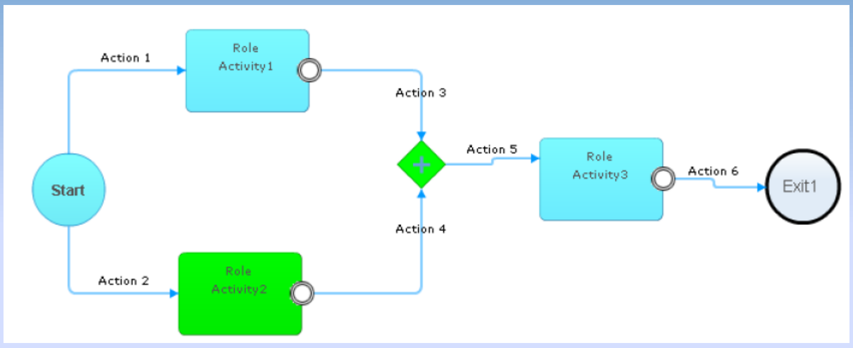


- After Recall



Recalling Task

- Recalling Parallel task

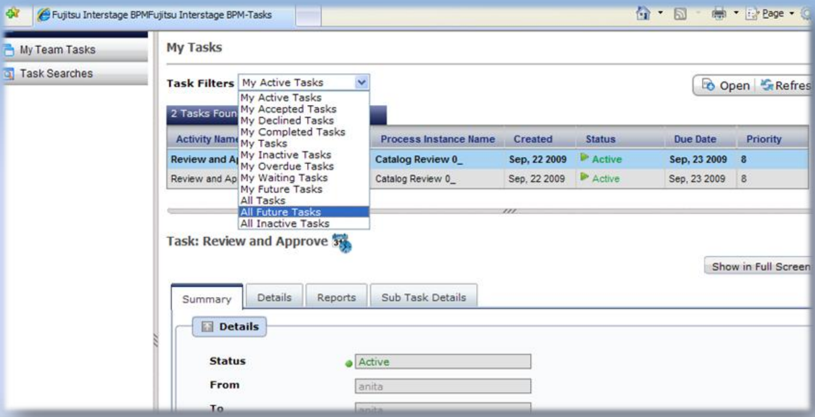


Future Tasks

- Allows users to estimate how many tasks might come to their queue in future.
- Helps in better planning future activities
- Users can search for future tasks in the task page on console.
- Future tasks show only incoming tasks for active instances/requests.

Future Tasks

- Users can filter the task on their "My Tasks" page in the Console to show likely future activities.



My Tasks

Task Filters: My Active Tasks

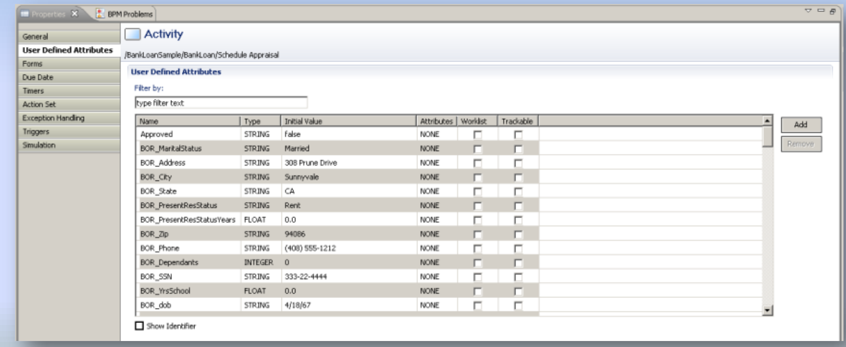
Activity Name	Process Instance Name	Created	Status	Due Date	Priority
Review and A	Catalog Review 0_	Sep, 22 2009	Active	Sep, 23 2009	8
Review and Ap	Catalog Review 0_	Sep, 22 2009	Active	Sep, 23 2009	8

Task: Review and Approve

Status	Active
From	anita
To	anita

Node Properties - User Defined Attributes

- User Defined Attributes – Process data variables.
- UDAs can be updated by user in the task details form:
  - STRING, FLOAT, INTEGER, LONG, BOOLEAN, DATE, BIGDECIMAL and XML



Activity

/BankLoanSample/BankLoan/Schedule Appraisal

User Defined Attributes

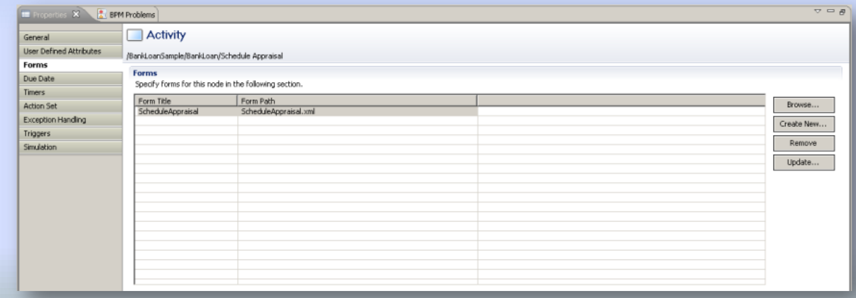
Filter by: type filter text

Name	Type	Initial Value	Attributes	Worklist	Trackable
Approved	STRING	false	NONE	☐	☐
BOR_MaritalStatus	STRING	Married	NONE	☐	☐
BOR_Address	STRING	308 Prune Drive	NONE	☐	☐
BOR_City	STRING	Sunnyvale	NONE	☐	☐
BOR_State	STRING	CA	NONE	☐	☐
BOR_PresentResStatus	STRING	Rent	NONE	☐	☐
BOR_PresentResStatusYears	FLOAT	0.0	NONE	☐	☐
BOR_Zip	STRING	94086	NONE	☐	☐
BOR_Phone	STRING	(408) 555-1212	NONE	☐	☐
BOR_Dependants	INTEGER	0	NONE	☐	☐
BOR_SSN	STRING	333-22-4444	NONE	☐	☐
BOR_YrsSchool	FLOAT	0.0	NONE	☐	☐
BOR_dob	STRING	4/18/67	NONE	☐	☐

☐ Show Identifier

Node Properties - Forms

- Forms can be attached to Activity nodes to present information to the users, required to complete the tasks.
- Users can edit/enter data and submit the task.
- Interstage BPM forms designer provides rich form building capability.
- JSP pages can also be attached as forms.



Activity

/BankLoanSample/BankLoan/Schedule Appraisal

Forms

Specify forms for this node in the following section.

Form Title	Form Path
ScheduleAppraisal	ScheduleAppraisal.xml

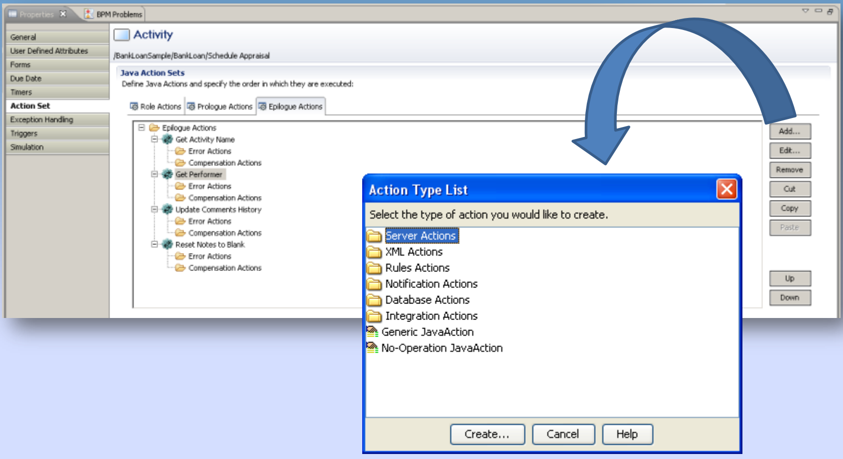
Java Actions

- Process Level
  - Init Action and Owner Actions
    - Executes at process instance start time
  - Commit Action
    - Executes upon process completion.
- Node Level
  - Role Action
    - Executed after assignee resolution and before task assignment
  - Prologue Action
    - Executed before the node performs its task
  - Epilogue Action
    - Executed after the node finishes its task and before the process instance moves on to another node

Pre-Defined Java Actions

- These groups of pre-defined Java Actions are available:
  - **Server** – enable to interact with the BPM Server
  - **XML** - add XML substructures, text elements, or attribute values to User Defined Attributes (UDAs) of type XML.
  - **Rules** - Java Action to invoke rules engines and execute business rules
  - **Notification** - notify users on events related to process execution
  - **Database** – allows to interact with external databases
  - **Integration** – allows to access external systems from process definition
  - **Custom** – Generic and No-operation Java Actions

Adding Actions



/BankLoanSample/BankLoan/Schedule Appraisal

Java Action Sets

Define Java Actions and specify the order in which they are executed:

```
Epilogue Actions
  Get Activity Name
    Error Actions
    Compensation Actions
  Get Performer
    Error Actions
    Compensation Actions
  Update Comments History
    Error Actions
    Compensation Actions
  Reset Notes to Blank
    Error Actions
    Compensation Actions
```

Action Type List

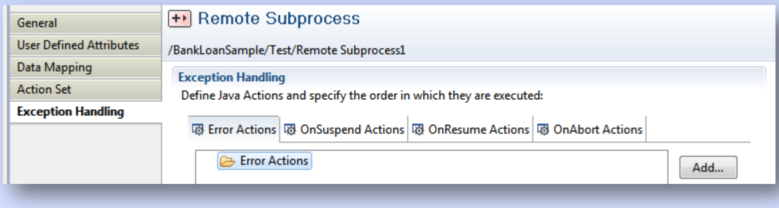
Select the type of action you would like to create.

- Server Actions
- XML Actions
- Rules Actions
- Notification Actions
- Database Actions
- Integration Actions
- Generic JavaAction
- No-Operation JavaAction

Exception Handling

FUJITSU

- Error Actions – Only available at Process Level
  - Used to handle error in "*Remote Subprocess Nodes*".
- OnSuspend, OnResume, and OnAbort
  - Executes when a process or task is Suspended, Resumed or Aborted.



Due Date and Timer

FUJITSU

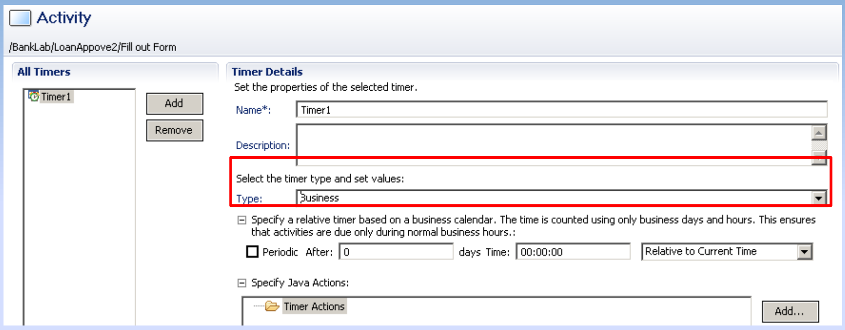
- Due Date
  - Specify when an activity is due to be completed once it has become active
  - Due date is displayed to the user and task is sorted in descending order by default.
  - Allows to take some action when due date expires.
- Timer
  - Allows to take some action at pre defined time/schedule
  - Can be repeated after each specified interval

Due Date and Timer Type

- Due Dates and Timer types depend on the schedule/date configuration
  - Absolute - triggers at a predefined specific date and time
  - Calendar - trigger after specified time has elapsed
    - Time specified in Days and Hours
  - Business - based on a Business Calendar
    - can use working hours from a Business Calendar
  - Advanced – define expressions using Time and Day codes and Business Calendar to define more complex timers
    - BM(-1); - the last business day of the month
    - WN(7); - the next Saturday after today
- Timers can be marked as Periodic if they are required to trigger repeatedly

Configure Timer

- Configure Timer to use business calendar



/BankLab/LoanAppove2/Fill out Form

Set the properties of the selected timer.

Name*: Timer1

Description:

Select the timer type and set values:

Type: Business

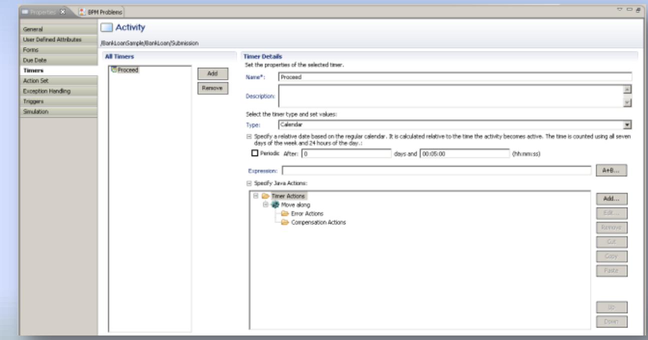
Specify a relative timer based on a business calendar. The time is counted using only business days and hours. This ensures that activities are due only during normal business hours.:

☐ Periodic After: 0 days Time: 00:00:00 Relative to Current Time

Specify Java Actions:

Due Date and Timer Actions

- Action can perform some task when due date or timer expires e.g.
  - Send an Email (escalation)
  - Evaluate a script or call an external interface
  - Escalate Task by re-assigning it to a new list of users.



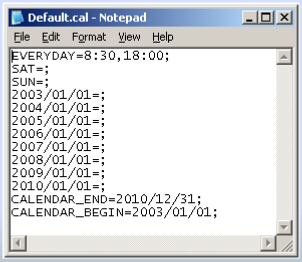
/BankLoanSample/BankLoan/Submission

All Timers: Proceed

Timer Details
Set the properties of the selected timer.
Name*: Proceed
Description:
Select the timer type and set values:
Type: Calendar
Specify a relative date based on the regular calendar. It is calculated relative to the time the activity becomes active. The time is counted using all seven days of the week and 24 hours of the day.:
Periodic After: 0 days and 00:05:00 (hh:mm:ss)
Expression:
Specify Java Actions:
Timer Actions
  Move along
    Error Actions
    Compensation Actions

Business Calendars

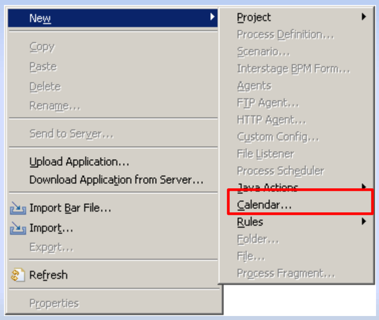
- Business Calendars define the hours of operation
- Used by Timers to calculate execution time
- Interstage Server has Default Calendar
  - Hours 8:30am – 6:00pm
  - Closed on Weekends
  - Closed Jan 1, 2003 - 2010
  - Valid for Jan 01, 2001 to Dec 12/31/10



```
EVERYDAY=8:30,18:00;
SAT=;
SUN=;
2003/01/01=;
2004/01/01=;
2005/01/01=;
2006/01/01=;
2007/01/01=;
2008/01/01=;
2009/01/01=;
2010/01/01=;
CALENDAR_END=2010/12/31;
CALENDAR_BEGIN=2003/01/01;
```

Creating Business Calendars

- Create Business Calendars
- Supports Multiple Calendars
- Calendars are Locale Specific



Defining Calendars

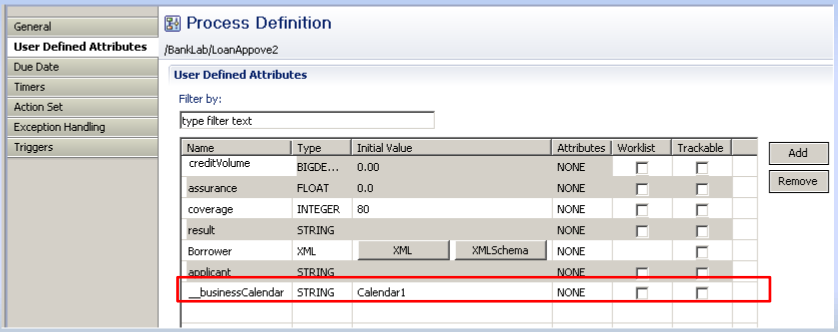
- EVERYDAY
  - Defines default business hours
    - Example: EVERYDAY=9:00,17:00;
- CALENDAR_END
  - End date the Calendar is valid until
  - Max10 years from CALENDAR_BEGIN
    - Example: CALENDAR_END=2010/12/31;
- CALENDAR_BEGIN
  - Begin date the Calendar is valid from
    - Example: CALENDAR_BEGIN=2010/01/01;
- TIMEZONE
  - The Time zone using GMT
    - Example: TIMEZONE=+10:00; (Brisbane Australia)
    - Example: TIMEZONE=-5:00; Eastern USA (New York)

Calendar: Optional Expressions

- <Day of Week>=[hours, hours];
  - Overrides EVERYDAY setting
  - SUN,MON,TUE,WED,THU,FRI,SAT
  - Hours of operation or blank no hours
  - Example: SAT=9:00,12:00;
  - Example: SUN=;
- <DATE> Specific Date
  - Overrides EVERYDAY and <DAY OF WEEK>
  - Example: 2010/12/25=;
  - Example: 2010/02/14=9:00,12:00; 15:30,17:00; (Long Lunch)
- DST Day Light Savings Time
  - Example: 2010/04/20=DST(1); Spring forward one hour
  - Example: 2010/10/19=DST(0); Fallback

Assigning Business Calendars

- Assign Business Calendar to BPD
- Create UDA __businessCalendar (double underscore)
  - Type String
- Assign value to name of calendar (no extension)
  - Example: Calendar1



Process Definition

/BankLab/LoanAppove2

User Defined Attributes

Filter by: type filter text

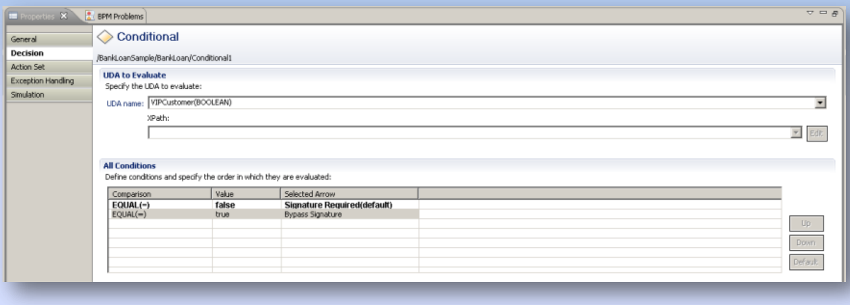
Name	Type	Initial Value	Attributes	Worklist	Trackable
creditVolume	BIGDE...	0.00	NONE	☐	☐
assurance	FLOAT	0.0	NONE	☐	☐
coverage	INTEGER	80	NONE	☐	☐
result	STRING		NONE	☐	☐
Borrower	XML	XML XMLSchema	NONE		☐
applicant	STRING		NONE	☐	☐
__businessCalendar	STRING	Calendar1	NONE	☐	☐

Additional Nodes

- Conditional and Complex Conditional Node
- Email Node
- Voting Node
- Sub Process Node
- Process Fragment
- Compound Node

Conditional Node Properties

- Used for conditional route of the process
- Decision can be based on a UDA value or *XPath expression (if UDA selected is XML type)*
- Supports simple *true-false* type conditions only



Conditional

/BankLoanSample/BankLoan/Conditional1

UDA to Evaluate

Specify the UDA to evaluate:

UDA name: VIPCustomer(BOOLEAN)

XPath:

All Conditions

Define conditions and specify the order in which they are evaluated:

Comparison	Value	Selected Arrow
EQUAL(=)	**false**	**Signature Required(default)**
EQUAL(=)	true	Bypass Signature

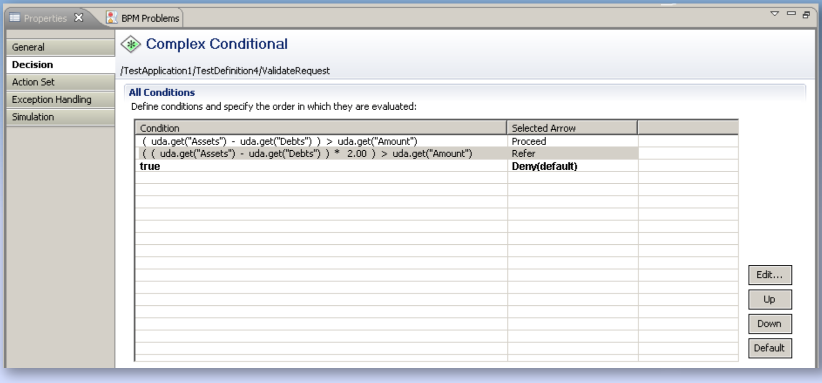
Complex Condition Node

FUJITSU

- Complex Conditional node supports complex expressions for conditional routing of process.
- Complex Conditional Nodes give you more flexibility as they allow to specify conditions combining **operators, dates, UDA's, constant values** etc.
- Easily create expressions using Expression Builder

Complex Condition Node

FUJITSU



Complex Conditional

/TestApplication1/TestDefinition4/ValidateRequest

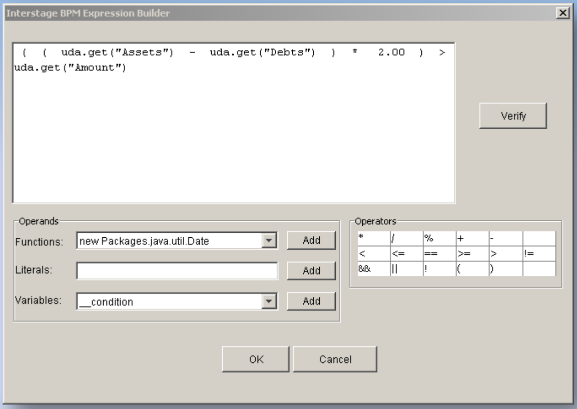
All Conditions

Define conditions and specify the order in which they are evaluated:

Condition	Selected Arrow	
(uda.get("Assets") - uda.get("Debts")) > uda.get("Amount")	Proceed	
((uda.get("Assets") - uda.get("Debts")) * 2.00) > uda.get("Amount")	Refer	
true	**Deny(default)**	

Expression Builder

- Build and verify complex conditions using a simple editor
- Also available for:
  - Java Actions
  - Triggers
  - Email Nodes



Interstage BPM Expression Builder

```
 (  (  uda.get("Assets")  -  uda.get("Debts")  )  *  2.00  )  >
uda.get("Amount")
```

Operands

Functions: new Packages.java.util.Date

Literals:

Variables: __condition

Operators

*	/	%	+	-	
<	<=	==	>=	>	!=
&&	||	!	(	)	

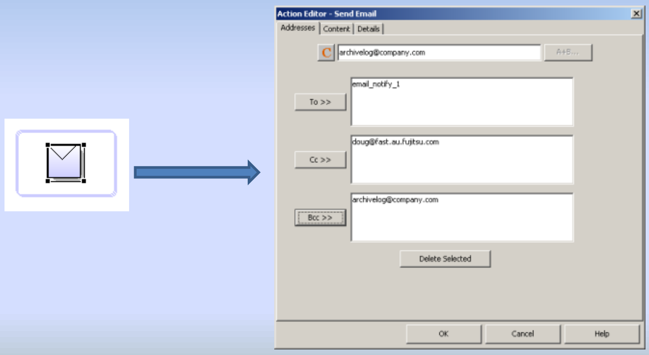
Email Node

- Use Email Node to send emails during process execution.
- After sending the email, Email Node activates all outgoing arrows.
- No user action is required with this node.

- The address, subject and body fields can either contain fixed text or a combination of text and UDA values.
- The Expression Builder is used to create complex expressions which combine UDA's, text and conditions.

Email Node

- Email Action can be attached to other nodes as well.
- Supports attachment



archivelog@company.com

To: email_notify_1

Cc: doug@fast.au.fujitsu.com

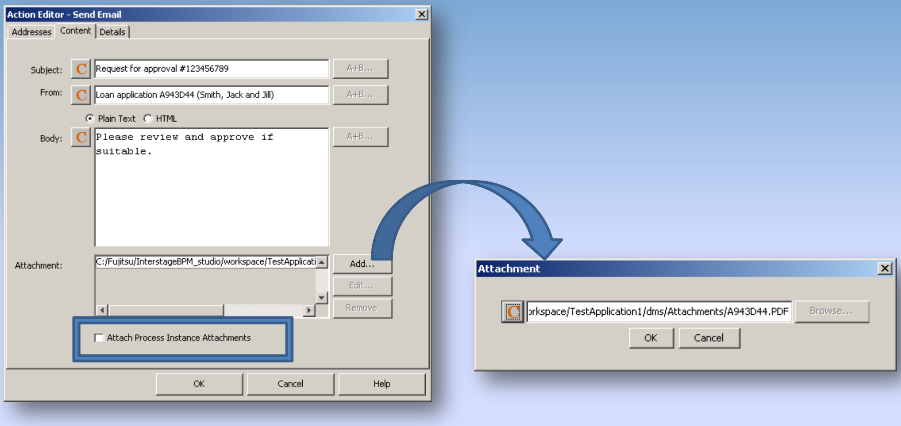
Bcc: archivelog@company.com

Email Node Properties

- Addresses
  - TO, CC, BCC
  - Supports email ids, UDAs or expressions
- Content
  - Subject, From, Body (email message)
  - Supports free text, UDA and expression
  - Supports Text or HTML content in body
- Attachment
  - Process attachments can be attached to email message
  - Document available in DMS can be attached
    - Use actual path, UDA or expression for document to be attached.

Email Node

FUJITSU


Action Editor - Send Email

Addresses | Content | Details

Subject: Request for approval #123456789

From: Loan application A943D44 (Smith, Jack and Jill)

(● Plain Text ○ HTML)

Body: Please review and approve if suitable.

Attachment: C:/Fujitsu/InterstageBPM_studio/workspace/TestApplicati

☐ Attach Process Instance Attachments

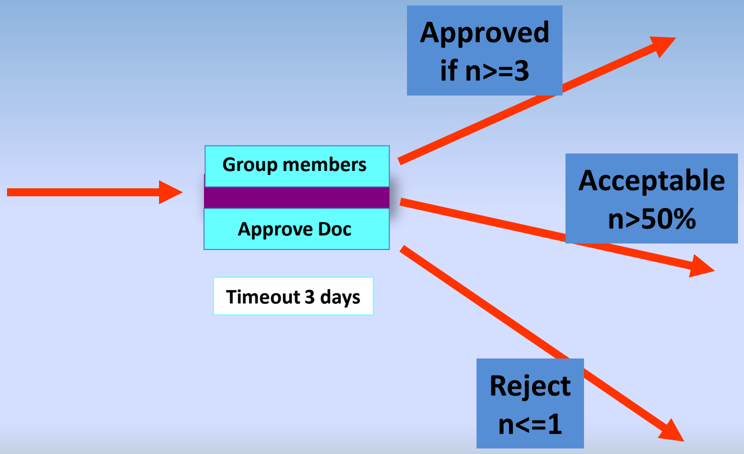
Attachment

rkspace/TestApplication1/dms/Attachments/A943D44.PDF

Voting Activity Node

FUJITSU

- Simplifies support for groups of people and common interaction patterns.



Voting Activity Node

FUJITSU

- A Voting Activity uses voting rules to determine the choice (outgoing arrow) that wins.
- Voting Activities are designed so that many assignees can make their own choice on a particular work item.

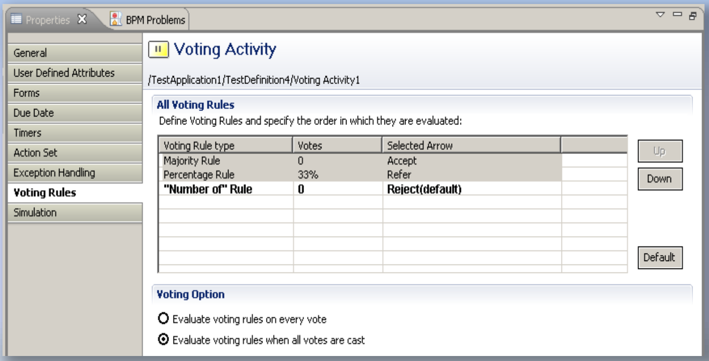
Voting Rules

FUJITSU

- Types of rules that can be assigned to Voting Activity Nodes:
  - Majority Rule
    - A majority of votes for that choice make it the winning choice.
  - Percentage Rule
    - The specified percentage of votes for that choice would make it the winning choice.
  - “Number of” Rule
    - The specified number of votes for that choice would make it the winning choice.
- When defining voting rules, you also choose a default rule that is chosen if none of the rules apply

Voting Activity Node Properties - Validation

- Evaluate voting rules on every vote
  - if you want to complete the activity as soon as a voting rule is satisfied.
- Evaluate voting rules when all votes are cast.
  - To make sure that everyone gets a chance to vote.



Voting Activity

/TestApplication1/TestDefinition4/Voting Activity1

All Voting Rules

Define Voting Rules and specify the order in which they are evaluated:

Voting Rule type	Votes	Selected Arrow
Majority Rule	0	Accept
Percentage Rule	33%	Refer
"Number of" Rule	**0**	**Reject(default)**

Voting Option

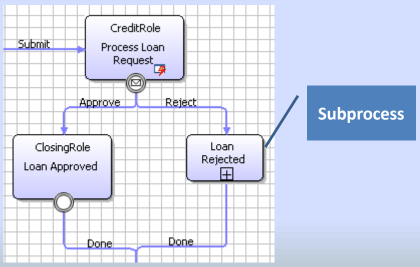
( ) Evaluate voting rules on every vote
[✓ Evaluate voting rules when all votes are cast]

Subprocess

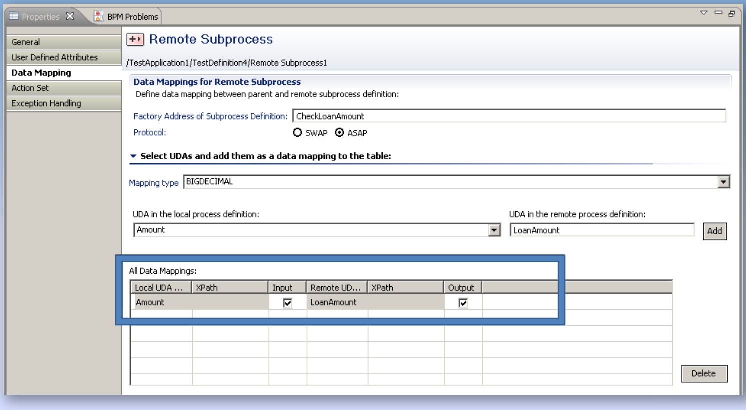
- Subprocess Node represents a step in a process where a task is accomplished by invoking another process.
- Invoked Subprocess should be a valid independent process definition with one start and at least one exit node.
- Usage
  - Promotes reusability
  - Helps improve visibility of large and complex process by breaking it down into smaller independent processes.
- Behavior:
  - instantiates a Process Instance from the Subprocess Definition
  - Data can be shared to and from Sub Process
  - Parent Process **waits for Subprocess to complete**

Subprocess –Data Mapping

- If UDA in parent process have same name as UDA in Subprocess, mapping is done automatically
- Modeler can map non-matching UDAs manually
  - UDA data-type needs to be same for mapping to work
- For XML UDAs, XPath can be used to mapping



Subprocess – Data Mapping



Remote Subprocess

/TestApplication1/TestDefinition4/Remote Subprocess1

Data Mappings for Remote Subprocess

Define data mapping between parent and remote subprocess definition:

Factory Address of Subprocess Definition: CheckLoanAmount

Protocol: SWAP [✓ ASAP]

Select UDAs and add them as a data mapping to the table:

Mapping type: BIGDECIMAL

UDA in the local process definition: Amount

UDA in the remote process definition: LoanAmount

All Data Mappings:

Local UDA ...	XPath	Input	Remote UD...	XPath	Output
Amount		✓	LoanAmount		✓

Process Fragments

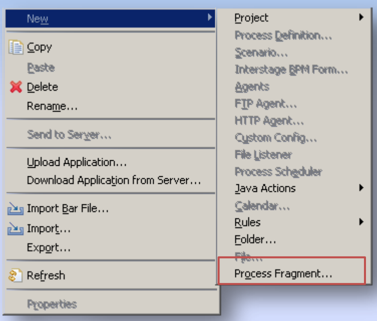
FUJITSU

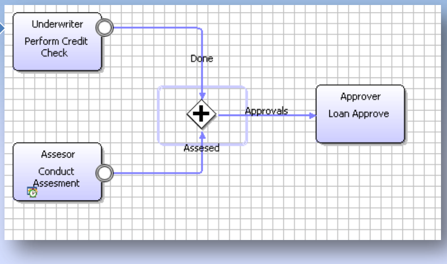
- Create re-usable process library.
- Fragments need not be valid process definitions.
- Any part of process can be saved as a fragment and used in other processes.
- UDAs are also stored in fragment and copied to the process definition.
- Edit fragments after copying if required
- Physically copied, not referenced.

Define Process Fragments

FUJITSU

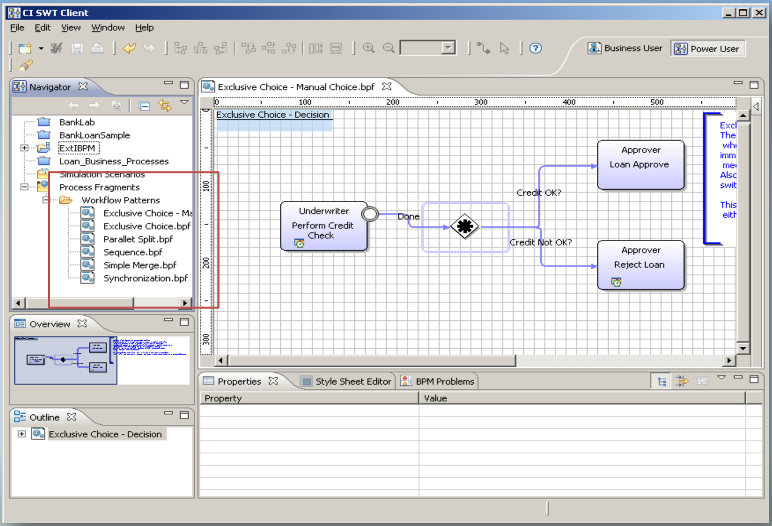
- Select parts of process to save as Fragment
- Create Folder to store
- Create Fragment





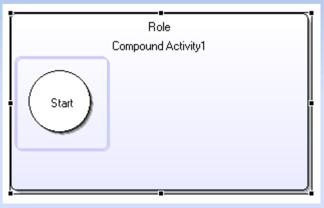
Process Fragments

- Drag and Drop to reuse



Compound Activity Node

- Provides option to group set of nodes in process
- Useful in modeling phases or milestones.
- Once a phase is complete, a milestone in the process can be said to be achieved and the process can move to the next activity or another phase



- Must have one start node and at least one exit node like a Subprocess.
- May have child nodes within compound node.

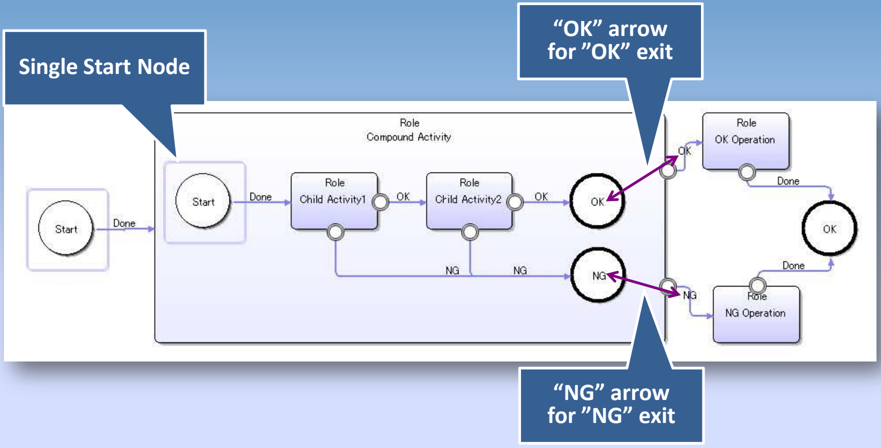
Compound Node Execution

FUJITSU

- when a compound node is reached in the process execution
  - Workitem is created for Compound Node and status is changed to "*WaitingOnSubProcess*"
  - Start node within compound node executes and control moves forward to next node.
  - When all child nodes complete execution, exit node executes.
  - Compound node status changes to "Completed"
  - Process moves forward.
- Compound node
  - May have due dates, actions etc like any activity node
  - Name of child "Exit" node MUST match the name of outgoing arrow from compound to next node in the process.
  - Cannot be recalled.

Compound Activity Node - Example

FUJITSU



shaping tomorrow with you