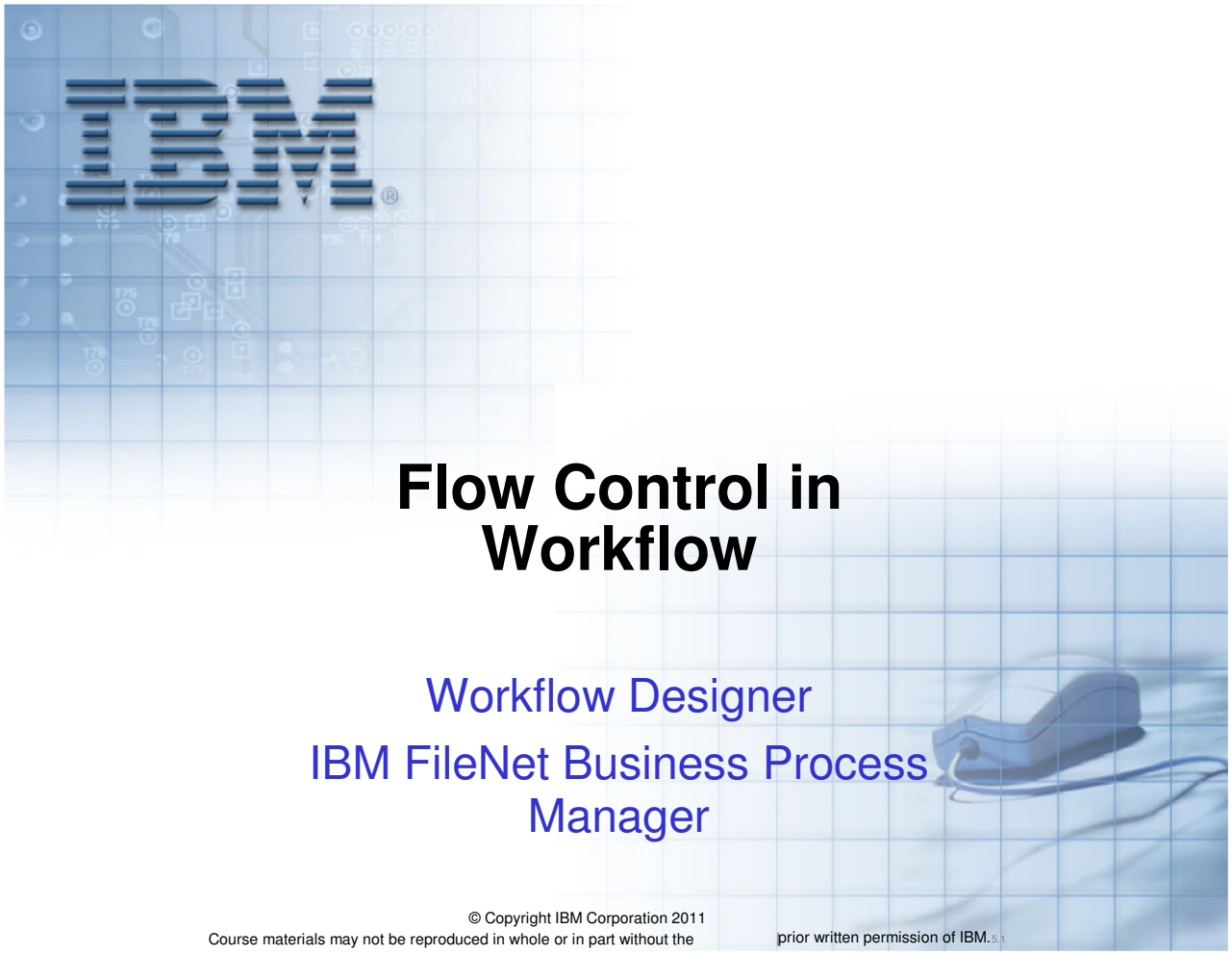


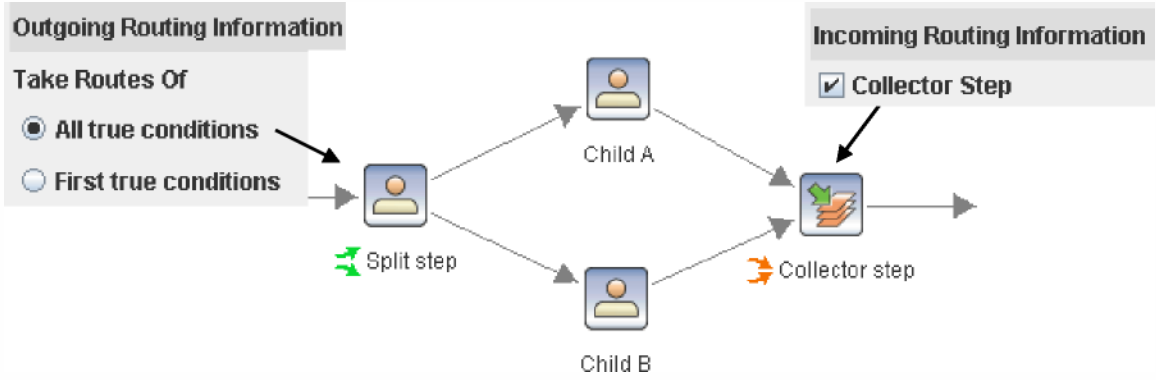
Flow Control in Workflow

Workflow Designer

IBM FileNet Business Process Manager

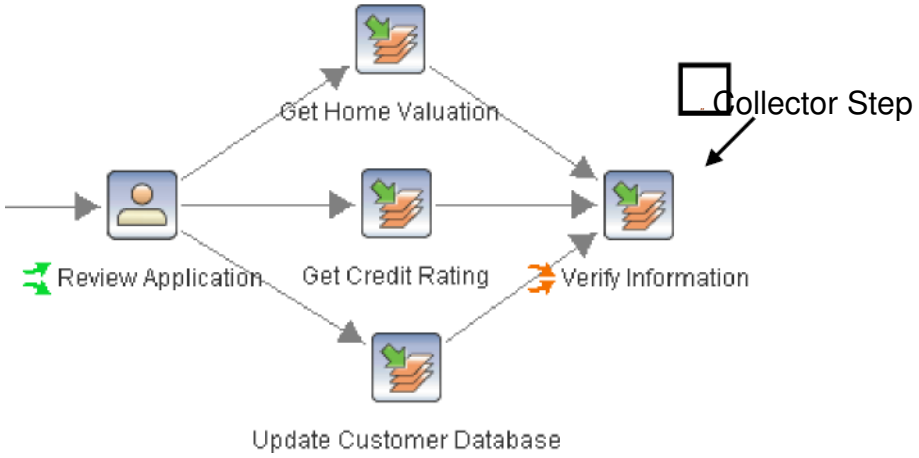
Modeling parallel execution

- Define an AND-split step
  - –On Step Properties > Routing tab, select the “All true conditions” option for outgoing routes.
  - –Create multiple true outgoing routes from the step.
- Define a collector step (also known as an AND-join step) –
  - Required to join multiple, parallel routes
  - –Joins the work back into a single path at end of all true routes



Example: Using parallel processing

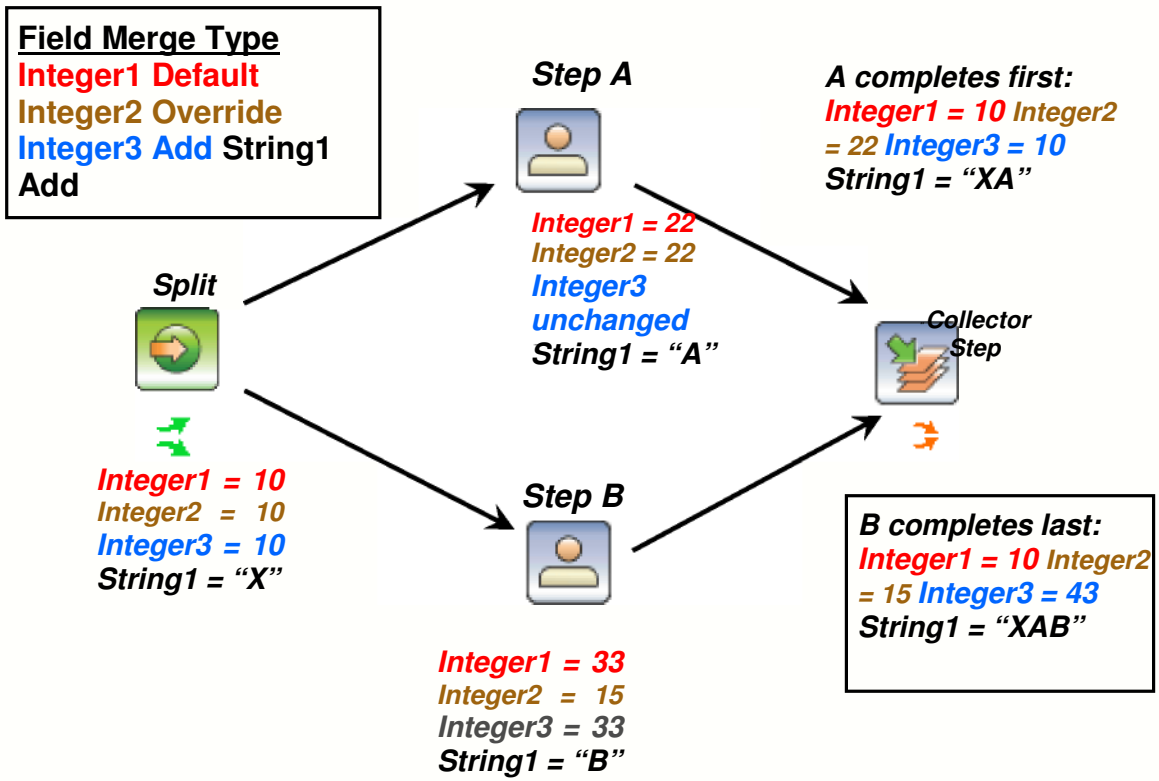
- The Review Application step specifies “All true conditions” for outgoing routes.
- Three steps are processed concurrently followed by the collector step.
- Three work items are created to follow each of the three paths.



Collection and data merge types

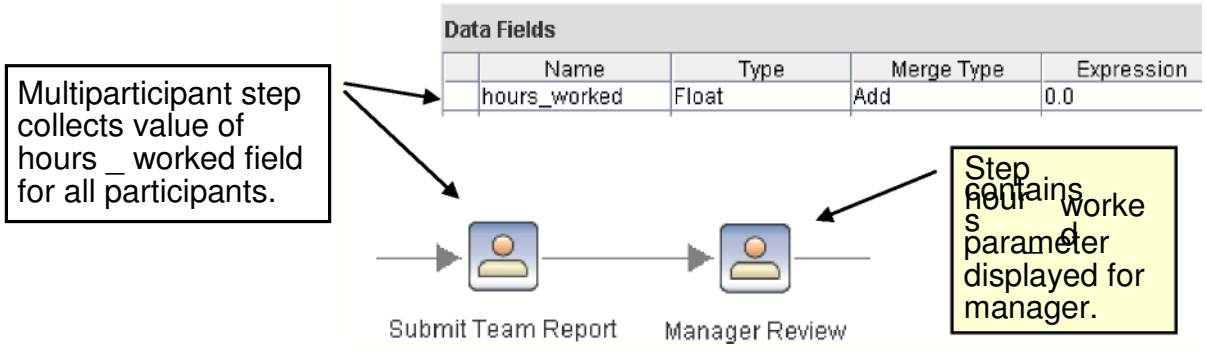
- Collection occurs in
  - –Collector step (AND-join step)
  - –Multiparticipant steps
- Collection is composed of the following:
  - –The parent work item waits in the Delay system queue until child processing is complete.
    - –Field merging of the child items' data fields to the parent work item
- Data merge type is defined in Workflow Properties for a data field and governs how the field merge is performed.
  - –Data merge types for simple data types: Default, Override, Add
  - –Data merge types for arrays: Default, Override, Override Entry, Append, Append No Dup, Add

How data merging works



Example: Data merging in a multiparticipant step

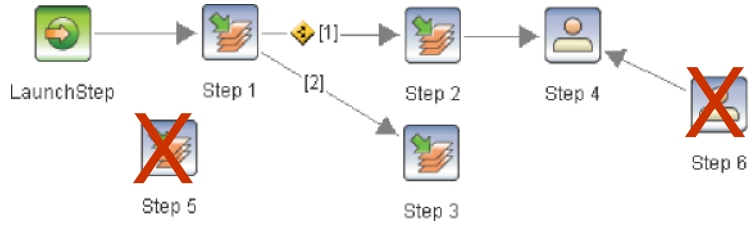
- Use case scenario –Manager of a project team wants to review total hours worked.
- Example solution
  - –Submit Team Report is a multiparticipant step completed by each team member.
  - –The hours_worked parameter is assigned a value by each team member.
  - –Float field uses Add merge type.



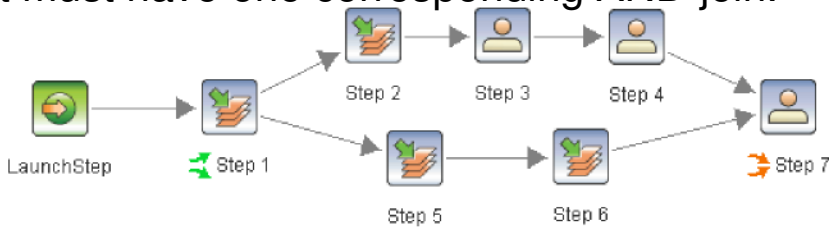
Parallel processing

Rules for creating valid workflow maps (1)

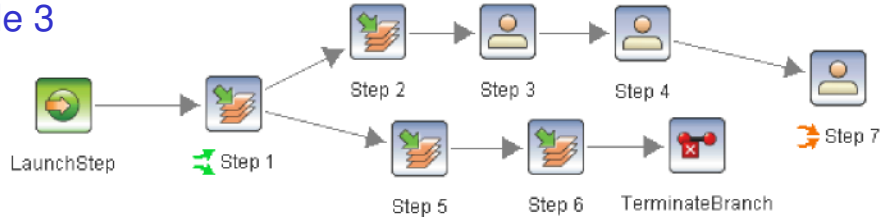
- All steps must be reachable from the Launch step.
  - Example 1



- AND-split must have one corresponding AND-join.
  - Example 2



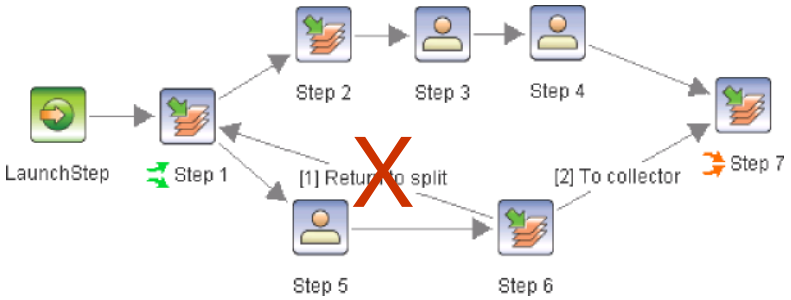
  - Example 3



Rules for creating valid workflow maps (2)

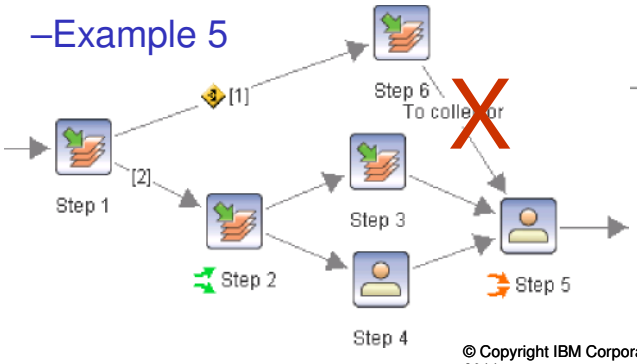
- Cannot return to AND-split without first going through collector

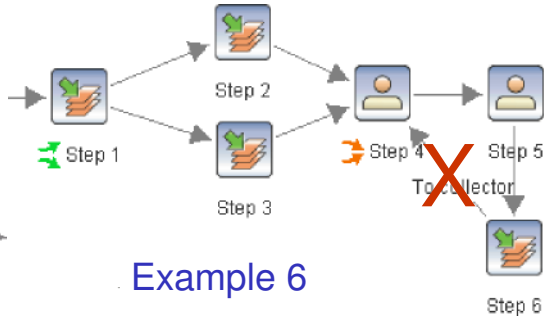
–Example 4



- Cannot go to AND-join without first going through AND-split

–Example 5





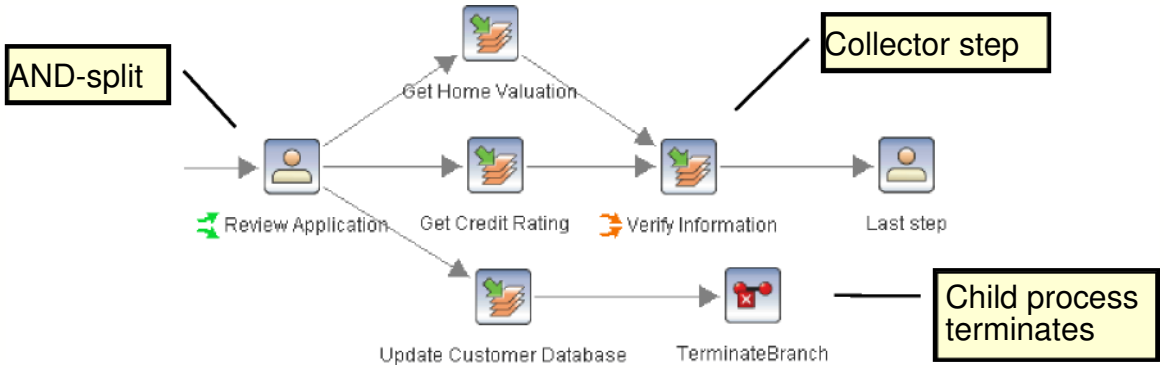
Example 6

Use of TerminateBranch system function

- Use case scenario
  - –In a workflow with an AND-split, the main process cannot wait for one of the branches of the split to complete.
- Design solution example
  - –Use TerminateBranch to end processing on one branch of the split.

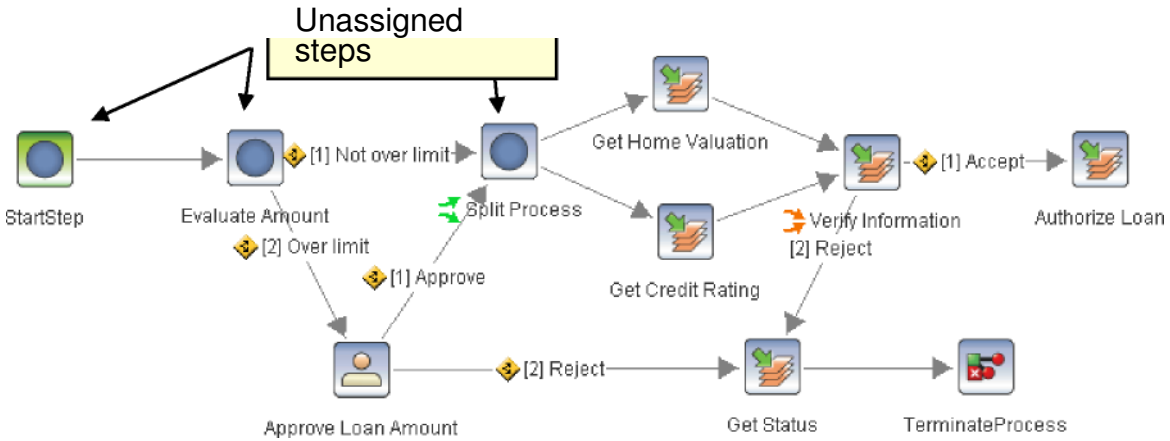
Result:

- Parent work item does not wait for Update Customer Database.
- Data fields are merged at the collector step and processing continues.



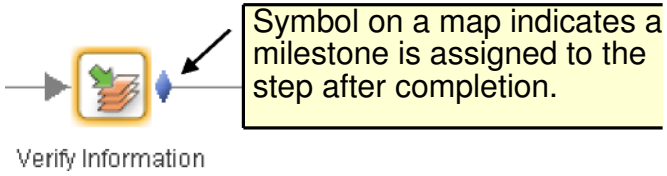
Use of an unassigned step

- Step that has an unspecified step destination
- Is not processed by a participant nor put in a work queue
- Can be used in some cases to facilitate flow
  - As a dummy step to evaluate routing conditions or perform calculations and assignments
  - As an AND-split step
  - As a placeholder for cycling back in a workflow



Milestones and workflow progress

- A milestone defines a notification point at a step in a workflow.
  - Before execution of a step
  - After completion of a step
- Used to trigger a message at a designated point
  - Message can be used to record progress of the workflow.
  - Message is displayed to workflow participants, trackers, and launch users.
  - Message is a string expression.
- Defined in Workflow Properties > Milestones tab
- Assigned to a step in Properties pane



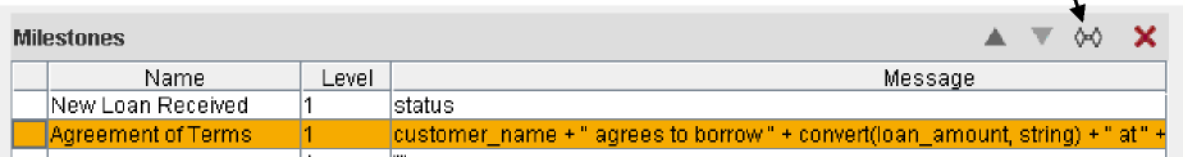
Milestone levels and messages

- Where a milestone and its message are displayed
  - –For a participant
- If available in the step processor, use Milestones view or Track Status action to display milestone information.
  - –For the originator who launched the workflow
- Use My Active Workflows view to display milestones for item.
- Use email, if email notification is configured. –For a tracker
- Use Milestones tab in Process Tracker to display messages.
- Use email, if email notification is configured.

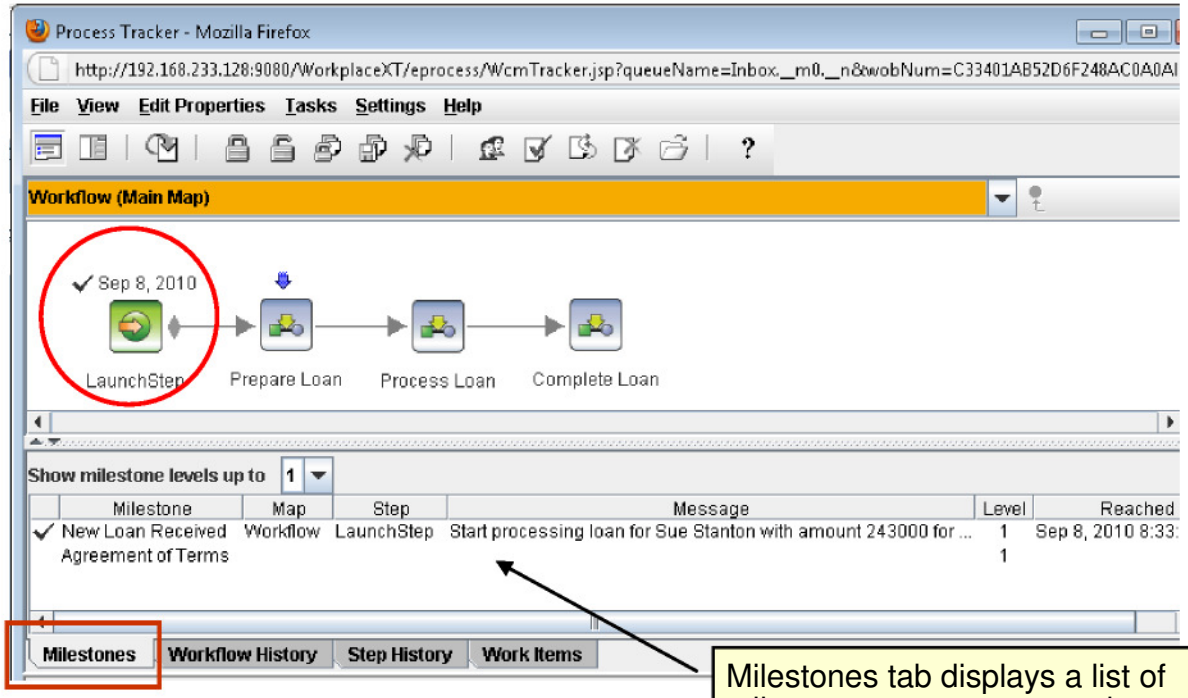
Define and use a milestone

1. Define the Milestone name, level, and expression in Workflow Properties Milestones tab.
   - Milestone name is displayed in Process Tracker and the step processor.
   - Milestone Level is used to determine the milestones displayed for a user.
1. Assign the milestone to one or more steps in the step Assignments tab.
   - Assignment Before Execution or Assignment After Completion

Click Milestone Usage icon to open a read-only display of milestone assignments by step.



Example: Milestone display in Process Tracker



Using event logs

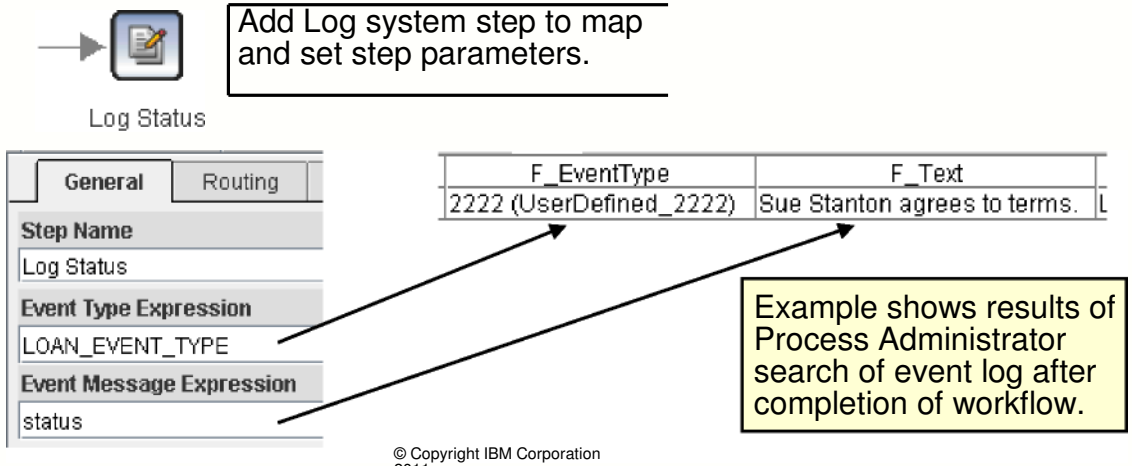
- Event logs keep a record of workflow history.
  - Contain a record of specific system- or workflow-related events used to track workflow activity
  - Are used in reports and tools to display workflow history
- At design time, you can specify when custom log entries are made in the event logs.
- After workflow processing completes, you can query event logs and produce reports on workflow activity that was logged.
  - Custom messages can provide meaningful information for your customized business reports.

Log system function

- Use to record a custom message in the event log.
- Specify two parameters:
  - –Event Type Expression
- An integer or integer expression greater than 1000
  - –Event Message Expression
- A string or string expression that is written to F_Text system field

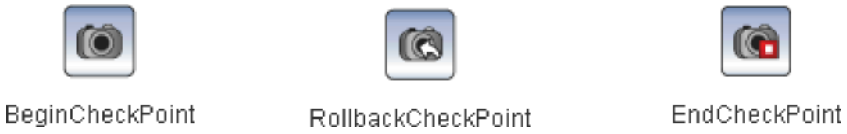
Example: Logging a data field

- Use case scenario –Management wants reports on status of loans (status data field).
- Design solution example
  –Record the value of the status data field in the event log by using the Log system function.
  –Develop a custom application to display management reports based on event logs using F_EventType and F_Text fields.



Rollback behavior in a workflow

- You can roll back work item data field values to the values held at a previous point in processing.
  - –You can specify which data field values are rolled back.
  - –You can specify whether work item processing resumes at that previous point or continues from the current point.
- Three system functions work together to enable checkpoint and rollback processing behavior:
  - BeginCheckPoint
  - RollbackCheckPoint
  - EndCheckPoint

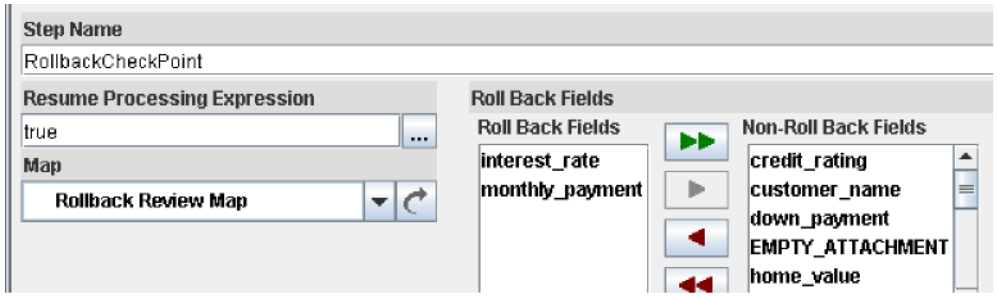


Checkpoint system functions

- BeginCheckPoint saves all data field values when the step is executed, for potential rollback to that point.
- When RollbackCheckPoint is executed:
  - –Specified stored data field values are recovered and replace the current field values.
  - –Work item processing moves back to the BeginCheckPoint or continues forward depending on the specified Resume Processing Expression.
  - –Optionally, a submap is executed before rolling back.
- EndCheckPoint discards the saved data field values. –Data fields retain current values.
  - –Processing continues with the next step.

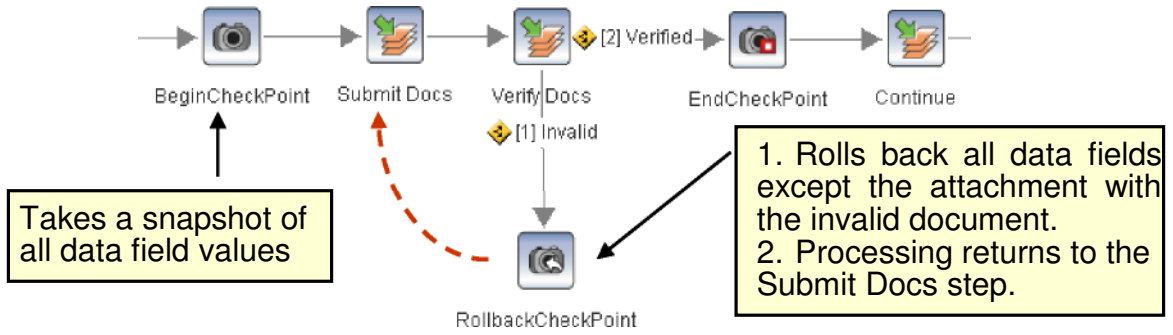
Specify where processing resumes

- For RollbackCheckPoint, you specify:
  - Resume Processing Expression
  - Optional submap
  - Rollback and non-rollback fields
- The Resume Processing Expression (Boolean expression) in RollbackCheckPoint properties is evaluated.
  - If true, then processing returns to the step after BeginCheckPoint.
  - If a submap is specified, then the submap is processed first.
  - If false, then processing continues to step after RollbackCheckPoint.



Example: Using a checkpoint

- Use case scenario
  - In a document verification process, you need the option to return an invalid document to the submitter and roll back some data values.
- Design solution example
  - BeginCheckPoint executes before the Submit Docs step to capture data values for potential rollback.
  - RollbackCheckPoint executes if documents must be resubmitted.
  - Resume Processing Expression evaluates to true and processing returns to the step following the BeginCheckPoint step.



Design considerations when using checkpoints

- Checkpoint blocks cannot be nested.
- A RollbackCheckPoint executed without an associated BeginCheckPoint calls the Malfunction system map.
- RollbackCheckPoint rolls back only the data fields. –Results from other system functions and tasks are not rolled back. –For example: Create system function creates a new work item.
- External events do not roll back.

Business process exceptions

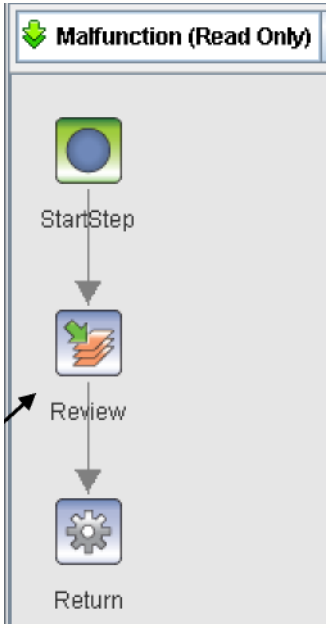
- When a problem occurs and processing cannot continue in a running workflow, a workflow exception occurs.
  - Work item is sent to the Malfunction system map.
  - These are business processing errors and user errors
  - By default, the work item is sent to the Conductor queue.
- You must design the workflow to handle business processing exceptions.
  - Work together with the application developer to identify potential exception cases and create custom solutions.
  - Avoid errors by using a combination of specialized error handling submaps and custom step processors.
  - Work items are then correctly processed rather than being sent to the Conductor queue.

Exception processing

Malfunction map and the Conductor queue

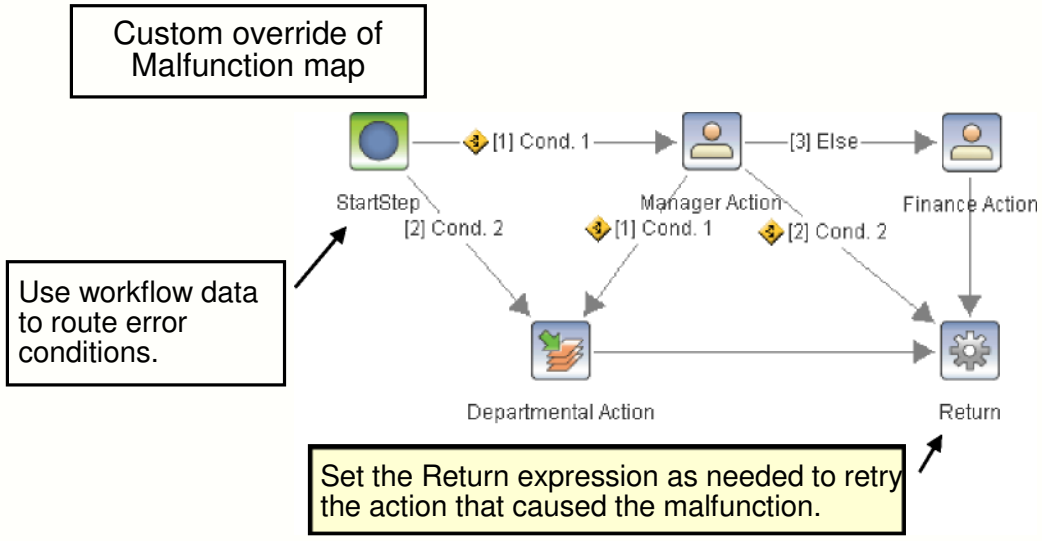
- Conductor queue
  - –A queue for work items in an exception state
  - –Default queue where a work item goes when a process exception occurs
- Items require administrative action.
  - –System cannot process work items in the Conductor queue without intervention.
  - –Use Process Administrator to access Conductor queue and view information about exceptions.

Conductor Review step is in the default Malfunction map, unless overridden.



Override the Malfunction system map

- You can override the default Malfunction map to trap workflow exceptions.
  - Work item returns to referring step after Malfunction submap completes.
  - If an exceptions occurs within overridden Malfunction submap, the work item goes to the Conductor queue.



Using Return in a Malfunction map

- Return system function in a called submap
  - –Return system function has a Boolean Return expression (also called the retry option).
  - –If the Return expression is true, then return to the calling map and repeat the action that called the submap.
  - –If the Return expression is false, then return to the calling map, skip the action that called the submap, and resume at the next state.
- The Return retry option is useful for exception handling.
  - –You can override the default Malfunction map to correct the error and return to the calling map in order to retry the step that failed.

Example: Workflow exception

- Malfunction occurs and work item goes to Conductor queue
  - View the information stack in Process Administrator.
  - Determine cause by reviewing the workflow map and data field values using Process Administrator and Process Tracker.
  - Example: No user is assigned to the step destination workflow group.

Information Stack:

Map	Step	Error Number	Error Message
Malfunction	Review Exception	0x0	
PrepareLoan	Confirm Information	0xd5240069	Malfunction: [Err=d5240069] Step: Invalid Participant. []

Information stack shows the step where the error occurred and the message.

- Possible design solutions
  - Use dynamic assignment of workflow groups and check for null or invalid users before the step that uses the workflow group.
  - Override Malfunction map and send work items to a workflow administrator who can assign a valid user.