

webMethods Integration Workshop – Day 2

February, 2016





Recap - Day

Learnt So Far (Recap & assessment)

- webMethods Datatypes
- Introduction to Services
- Documents
- Flow Services
- Flow Steps MAP
- Flow Steps INVOKE
- Flow Steps BRANCH





Introduction

- This course will help participants to simulate the basic FlowSteps (Sequence, Loop, Repeat and Exit)
- Participants will get hands on basic coding standards during Integration Development.



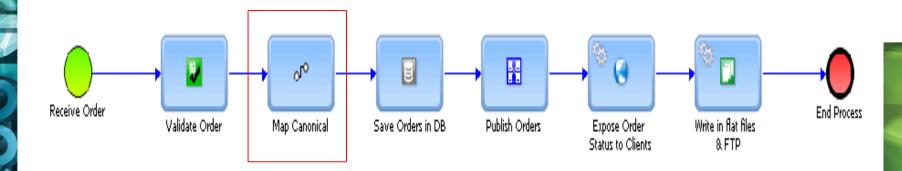


Objectives

- Of Get hands on experience in creation of flow service
- Basic coding standards, i/o signature.
- To know about basic flow steps SEQUENCE, LOOP, REPEAT and EXIT.



webMethods Pilot Project Progress



Outcome of this course:

Trainees should be able to create a flow service (for Mapping) which does the mapping of the OrderCustomer information as part of Practical session



Software versions

This class focuses on the webMethods suite

webMethods Integration Server

Broker/ UM

Software AG Designer



Chapters

Day 3

Flow Step - SEQUENCE

Flow Step - LOOP

FlowStep - REPEAT

FlowStep - EXIT

FlowStep – Basic Coding standards





Flow Step - SEQUENCE

- You use the SEQUENCE step to build a set of steps that you want to treat as a group.
- Steps in a group are executed in order, one after another. By default, all steps in a flow service, except for children of a BRANCH step, are executed as though they were members of an implicit SEQUENCE step (that is, they execute in order, one after another).

The most common reasons to do this are:

- To group a set of steps as a single alternative beneath a BRANCH step.
- To specify the conditions under which the server will exit a sequence of steps without executing the entire set.
- In an implicit sequence, when a step fails, the server automatically exits the sequence (that is, the **Exit on property is set to FAILURE).**





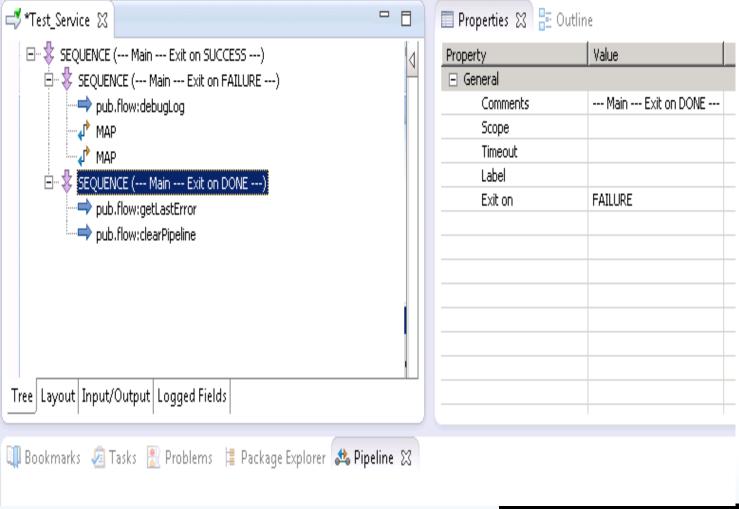
SEQUENCE – Exit on Condition

© SUCCESS -- Exit the sequence when any step in the sequence succeeds. When a SEQUENCE exits under this condition, the server considers the SEQUENCE step successful, even if all its children fail.

ODONE -- Execute every step in the sequence even if one of the steps in the sequence fails. The server considers a SEQUENCE step successful as long as it executes all of its children within the specified time-out limit.



SEQUENCE – Main Try Catch Block







- The LOOP step repeats a sequence of child steps once for each element in an array that you specify. For example, if your pipeline contains an array of purchase-order line items, you could use a LOOP step to process each line item in the array.
- To specify the sequence of steps that make up the body of the loop (that is, the set of steps you want the LOOP to repeat), you indent those steps beneath the LOOP as shown in the following example.
- Simple Loop step

```
*Loop_Service \( \text{S} \)

MAP

pub.flow:debugLog

LOOP over 'Emp_Name'

MAP

MAP

MAP

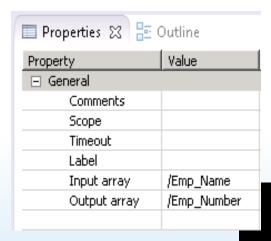
SEQUENCE

Tree Layout Input/Output Logged Fields
```





- Property Input Array The LOOP step requires you to specify an input array that contains the individual elements that will be used as input to one or more steps in the LOOP.
- At run time, the LOOP step executes one pass of the loop for each member in the specified array.
- For example, if you want to execute a LOOP for each line item stored in a purchase order, you would use the document list in which the order's line items are stored as the LOOP's input array.
- The array you specify can be any of the following data types:
 - String list
 - String table
 - Document list
 - Object list





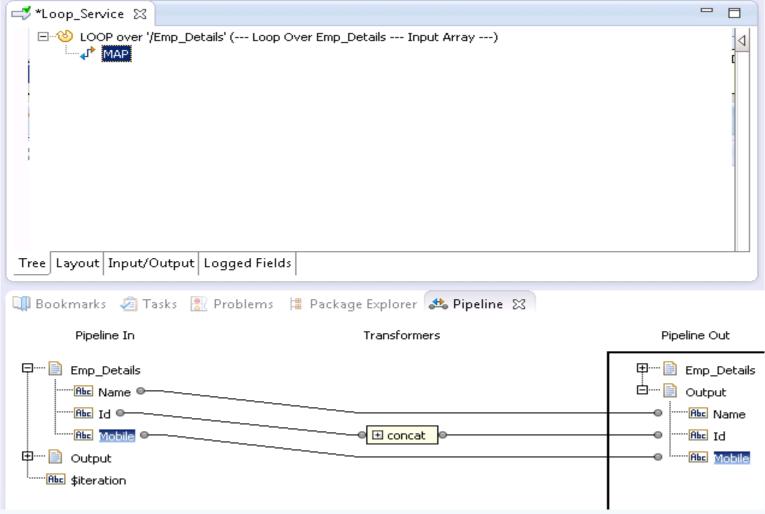
LOOP - Collecting output - OutputArray

- Property Output Array If your LOOP step produces an output variable, the server can collect that output into an array in the pipeline.
- To do this, you use the Output array parameter to specify the name of the array variable into which you want the server to collect output for each iteration of the loop.
- For example, if your loop checks inventory status of each line item in a purchase order and produces a String called InventoryStatus each time it executes, you would specify InventoryStatus as the value of Output array.
- To collect output from each pass of the loop, specify the name of the output variable that you want the server to collect for each iteration.

■ Properties 🛭 📴 Outline	
Property	Value
□ General	
Comments	
Scope	
Timeout	
Label	
Input array	/Emp_Name
Output array	/Emp_Number

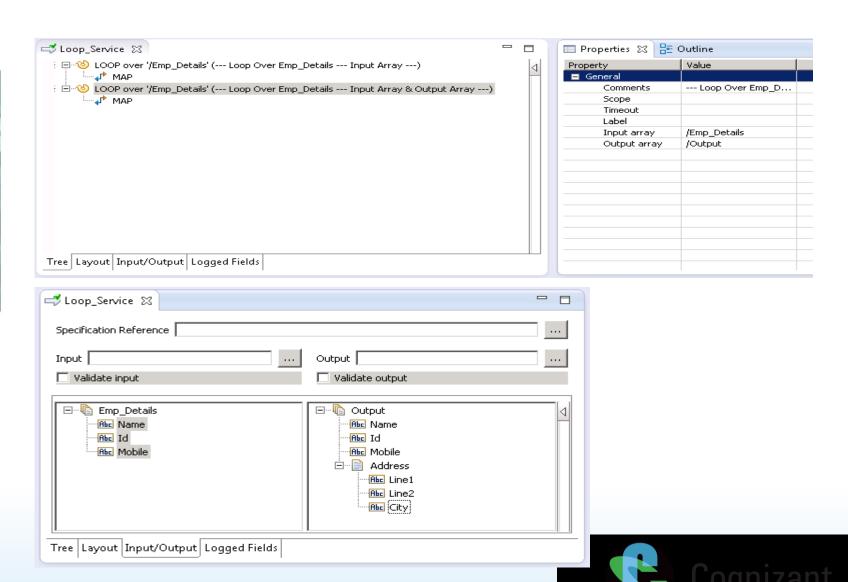


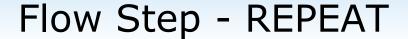
LOOP - Pipeline view (Input Array)



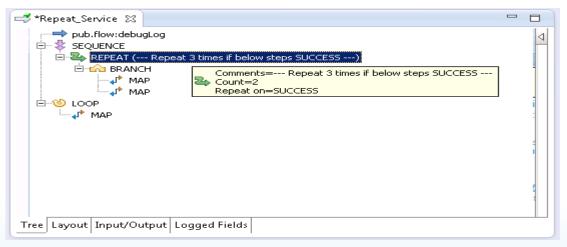


LOOP - Pipeline view - Output Array





- The REPEAT step allows you to conditionally repeat a sequence of child steps based on the success or failure of those steps.
- Re-execute (retry) a set of steps if any step within the set fails. This option is useful to accommodate transient failures that might occur when accessing an external system.
- Re-execute a set of steps until one of the steps within the set fails. This option is useful for repeating a process as long as a particular set of circumstances exists.

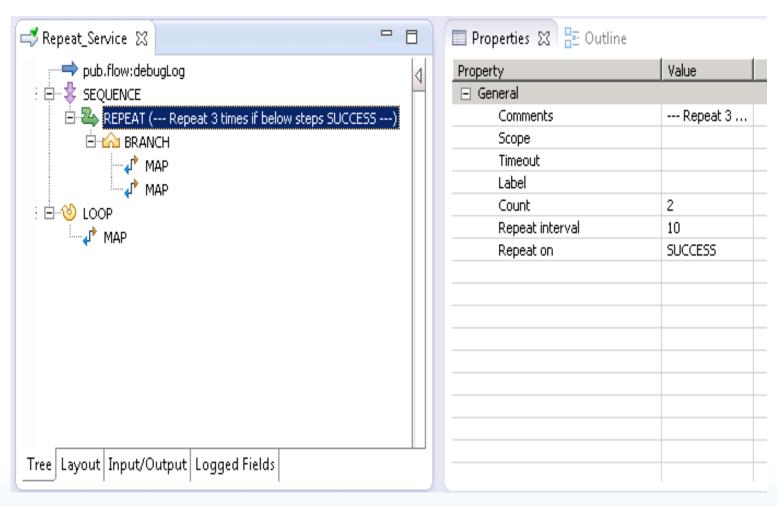






- <u>"Repeat On"</u> When you build a REPEAT step, you set the Repeat on property to specify the condition (success or failure) that will cause its children to re-execute at run time.
- FAILURE -- Re-executes the set of child steps if any step in the set fails.
- SUCCESS -- Re-executes the set of child steps if all steps in the set complete successfully.
- <u>"Repeat Count"</u> The REPEAT step's Count property specifies the maximum number of times the server re-executes the child steps in the REPEAT step..
- Does not re-execute children.
- -1 or blank Re-executes children as long as the specified Repeat on condition is true.

REPEAT – Pipeline view







Flow Step - EXIT

- The EXIT flow step allows you to exit the entire flow service or a single flow step. You specify whether you want to exit from.
- When you use the EXIT step, you indicate whether exiting should return a successful condition or a failure condition.

Examples of when to use the EXIT step include to:

- Exit an entire flow service from within a series of deeply nested steps.
- Throw an exception when you exit a flow or a flow step without having to write a Java service to call Service.throwError().
- Exit a LOOP or REPEAT flow step without throwing an exception.



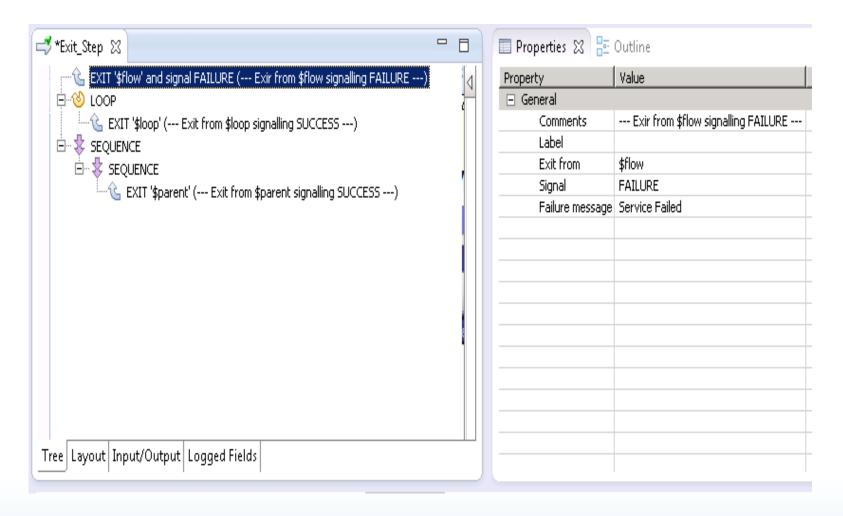


EXIT - Properties

- <u>"Exit from " -- The flow step from which you want to exit. Specify one of the following:</u>
 - \$loop -- Nearest ancestor LOOP or REPEAT flow step
 - \$parent -- Parent flow step, regardless of the type of step
 - \$flow -- Entire flow service.
- "Label" -- Nearest ancestor flow step that has a label that matches this value.
- "Signal" --Whether the exit is to be considered a success or a failure. Specify one of the following:
 - SUCCESS -- Exit the flow service or flow step with a success condition.
 - FAILURE -- Exit the flow service or flow step with a failure condition. An exception is thrown after the exit. You specify the error message with the Failure message property.



EXIT- Pipeline view







Basic Coding Standards

- Package names should always be capitalized. If a name is composed of many words, the first letter of each word must be capitalized.
- webMethods node objects cannot contain:
 - Reserved words and characters that are used in Java or C/C++ (such as for, while, and if)
 - Digits as their first character & Spaces
 - Control characters and special characters like periods (.), including:

- © Create only one top-level folder in each package.
- No folder should be created outside the package base folder
- Service names should start with a verb, in lowercase, and any subsequent objects in a service name should be capitalized (Examples: postNewHireToPeopleSoft, writeFile)



Summary

What have we learnt today?

- Flow Steps SEQUENCE
- Flow Steps LOOP
- Flow Steps REPEAT
- Flow Step EXIT
- Basic Coding Standards





- ❖ What is main property name to be set for SEQUENCE Flow step ? What are the options available for the property ?
- ❖What does Exit from \$parent means in EXIT flow step?
- What is the property value set for Catch block? Why?
- ❖ What is Output array? When do we use it?
- What DataTypes can be specified in InputArray of Loop?
- What happens if Repeat count is set as 0 for REPEAT step?



Thank you

