How To... Configure Inbound Processing in ccBPM Part I: Delivery Mode
Version 1.00 - December 2006

Applicable Releases:
SAP NetWeaver ’04 Support Package Stack 17
1 Scenario

Starting with SP17 of SAP NetWeaver '04 (SP8 of SAP NetWeaver '04s respectively) inbound processing of XI's cross-component Business Process Management (ccBPM) can be configured using transaction SWF_INB_CONF. This How-to Guide starts a series of documents providing decision criteria how to set the relevant inbound processing parameters properly based on your process definition.

In part I of the series, the delivery mode is discussed. The screenshots are based on SP17 and SP8 respectively. The following parts will deal with queue and transaction handling of ccBPM.

2 Introduction

Inbound processing is a runtime component of XI's ccBPM that links the Integration Engine to the Business Process Engine. It defines the way how XI messages are handed over to receive steps of process instances. The messages can thereby either start a new process instance or can be bound to a running process instance using a correlation.

Within this How-to Guide we will first discuss the architectural principles of inbound processing. In a second step we will highlight the impact of the delivery mode parameter on the business process runtime based on selected process fragments. Finally the necessary steps to assign the delivery mode to an Integration Process Service are presented.

3 Delivery Mode

The runtime of the Business Process Engine supports three potential delivery modes: "Without Buffering", "Buffering Possible" and "Without Buffering (Obsolete)". We generally recommend setting the delivery mode to "Without Buffering". On the one hand side, direct delivery (i.e. without buffering) increases your processing performance. On the other hand it provides a more robust handover of messages to a running process instance. Nevertheless there might be cases where buffering is mandatory. This guide aims to provide decision criteria which delivery mode is adapted for specific process definitions.

Note: Delivery mode "Without Buffering (Obsolete)" is available for compatibility reasons. Please do not select this delivery mode, except when you’re explicitly asked by SAP's support organization to do so.

3.1 Classic Setting: Inbound Processing with Buffering

3.1.1 Runtime Behavior

If you choose the delivery mode "Buffering Possible" using transaction SWF_INB_CONF, all messages that fit to an active correlation are taken up by a running process instance. They are associated to a running process instance, regardless if there is or will be an active receive step in this process instance. The messages are buffered if the dedicated process instance does not provide an active receive step at the actual stage of your process execution. The message will be processed by the process instance as soon as the corresponding receive step inside your process definition is encountered. If multiple messages for one receive step have been buffered they are processed according to the sequence they entered the buffer. If the running process instance provides a corresponding active receive step for an incoming message, the message is delivered directly without any buffering.
Figure 1: Delivery mode with buffering. Messages dedicated to a specific process instance are stored intermediately if the process instance does not provide an open receive step.

The usage of inbound processing with buffering may impact the delivery of messages the following way:

- Even if the process instance does not provide a receive step at the present stage of the execution, the message might be buffered and delivered later to the process instance.
- If a process instance has long running open correlations, messages might be delivered to the process instance being buffered even though the process instance does not provide any further receive step. In this case, if the relevant process has not been carefully designed by the developer, messages might not be delivered anymore. Messages not being delivered to a running process instance can be monitored in CCMS (Unprocessed XML Messages). Narrow down the duration of correlations to the strict minimum is a good development practice.

SAP recommends using buffering only in limited and well-defined scenarios which will be discussed in the next section.

3.1.2 Use cases

There might be situations when buffering is mandatory. If the process definition is not subject to change, the following cases require inbound processing with buffering:

1. Interrupting synchronous processing in collect scenarios

Since the synchronous processing scheme is interrupted by the transformation step, there might be situations when the process instance does not provide an open receive step for incoming messages. Therefore, if the process definition cannot be changed, buffering is mandatory in this scenario. Process steps interrupting synchronous process execution are transformation steps, receive steps, send steps, wait steps, receiver determination steps, and control steps throwing alerts.
2. Interrupting synchronous processing in request/response scenarios

For the same reason buffering is mandatory, if your process definition requires a transformation step, receive step, send step, wait step, receiver determination step, or control step throwing alerts between a send and receive step, spanning an asynchronous request/response scenario.

3. Sequential receive steps in a process definition

If the sending application cannot guarantee the sequence of XI messages, a deadlock situation can be induced without buffering. In case the message for receive step Rec2 is transferred to the process type specific queue (XBQO$PE_WS...) before the message intended for Rec1, it prevents the latter from being delivered and hence the subsequent processing is blocked.

3.2 Inbound Processing Without Buffering

3.2.1 Runtime Behavior

If buffering is deactivated, a message is taken up by a running process instance, if it fits to the open correlation and only if the process instance provides an open receive step at that stage of the process. The messages are handed over to the running process instance directly. Therefore the developer has to ensure that the process instance provides an open receive step for any message dedicated to that instance. This can be achieved following the design patterns presented in section 3.2.2.

In other words, if a message is to be sent to a process instance, the relevant instance has already to be waiting for this incoming message with an open receive step. If this is not the case, the process type specific queue (XBQO$PE_WS...) goes to status "SYSFAIL" with error message "No active correlation for message...".
Figure 2: Delivery mode without buffering. Messages dedicated to a specific process instance are delivered directly to the process instance which has to provide an open receive step.

Most scenarios requiring inbound processing with buffering (see section 3.1.2) can be changed slightly, so buffering is not mandatory anymore.

3.2.2 Use cases
In the following section exposes techniques to guarantee that process instances provide open receive steps.

1. Keep synchronous processing in collect scenarios

If the transformation step is removed from the loop, the synchronous processing is continued until the next receive step is encountered. In addition, the correlation should be defined locally within the block enveloping the loop. This guarantees that the process instance always provides an open receive step for a given correlation.

2. Keep synchronous processing in request/response scenarios

In asynchronous request/response scenarios, the sequence of send and receive step must not be interrupted by any transformation step, receive step, send step, wait step, receiver determination step, or control step throwing alerts.
3. Parallel receive steps in a process definition

If the sending application cannot guarantee the sequence of XI messages, the corresponding receive steps have to be arranged in a fork. In this case either the receive step on the upper or the one on the lower branch will be executed first and the sequence of the incoming XI messages is not relevant anymore.

3.3 Assigning the Delivery Mode
The required delivery mode has to be assigned to the relevant task representing the runtime object of the Integration Process Service.

1. The task which represents the runtime component of the Integration Process Service defined in the Integration Directory and can be retrieved using transaction SXI_CACHE. Select “Integration Processes”. 

![Diagram showing parallel receive steps and fork]

If the sending application cannot guarantee the sequence of XI messages, the corresponding receive steps have to be arranged in a fork. In this case either the receive step on the upper or the one on the lower branch will be executed first and the sequence of the incoming XI messages is not relevant anymore.
2. Search for the name of your Integration Process set in the Integration Repository or the name of the service, set in the Integration Directory to retrieve the name of the task.

3. Using transaction SWF_INB_CONF, the delivery mode of the task can then be set to the following values: “Without Buffering”, “Buffering Possible” and “Without Buffering (Obsolete)” in transaction SWF_INB_CONF. Note that these settings are not transported and that they have to be set up in each system individually.