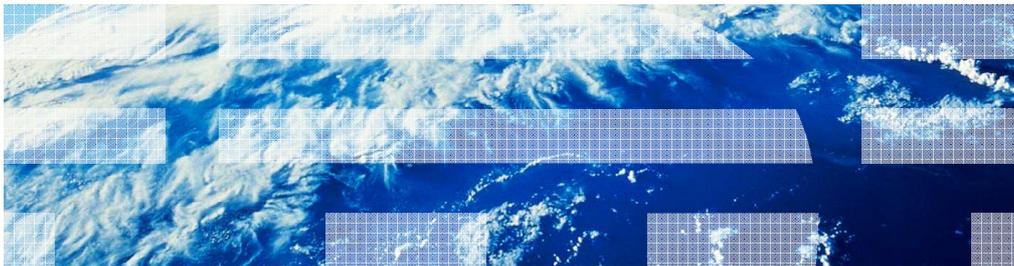


---

WebSphere Process Server V7  
WebSphere Integration Developer V7  
WebSphere Enterprise Service Bus V7

Store and forward feature



© 2010 IBM Corporation

This presentation will provide details on the store and forward feature, and how you can configure it using WebSphere® Integration Developer

## Goals

- Introduce the store and forward concepts
- Scenarios that are supported and not supported
- how to configure store and forward
- business space widgets

The goals for this presentation are to introduce the store and forward concepts and what scenarios are supported or not supported. How to configure the store and forward feature in your scenario and the business space widgets you can make use of for this feature are also covered

## What is store and forward?

- When a target service is unavailable, you can
  - setup WebSphere process server and Websphere Enterprise Service Bus to store the messages that are sent to target service
  - forward the saved messages once the service is available.
- Store and forward available
  - for target services that are called asynchronously and synchronously
  - Store and forward qualifier can be setup on asynchronous services only

Store and forward is a new feature introduced in version 7.0 . This feature allows you to configure a store where the system will start storing the messages if the target service being called is unavailable instead of generating failed events. Generating failed events for each failed request involves a round trip time in the system, and then you have to manually forward every failed message from the failed event manager. The advantage with the store and forward feature is that only the first request for the service will result in failed event. Any subsequent requests will be stored in a queue. Once the service is available, you can start “forward”, which will initiate the process of sending all the messages out of the queue. You do not need to forward each message individually. If there was any dependency on the first failed event generated with any of the messages in the queue, you need to go to failed event manager and resubmit the message. Only then will the subsequent messages from the store and forward queue will be sent out.

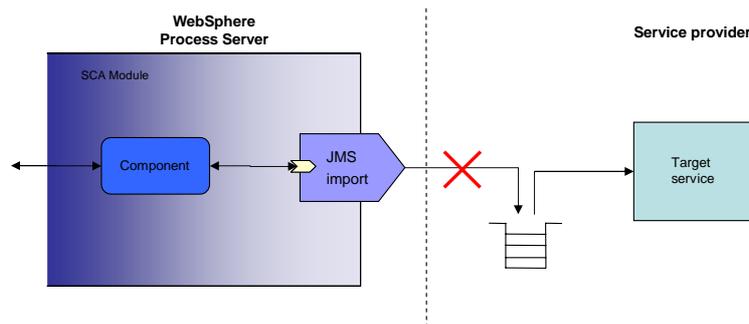
You can setup store and forward for any type of services, either synchronous or asynchronous. The only limitation is that you have to set the store and forward qualifier at the asynchronous invocation point in the flow.

***Store and forward  
What scenarios are supported?***

The next set of slides covers what scenarios are supported for store and forward.

## Scenario 1- Asynchronous imports – JMS, MQ, etc

- import unable to make a connection to the external destination for example queue



5

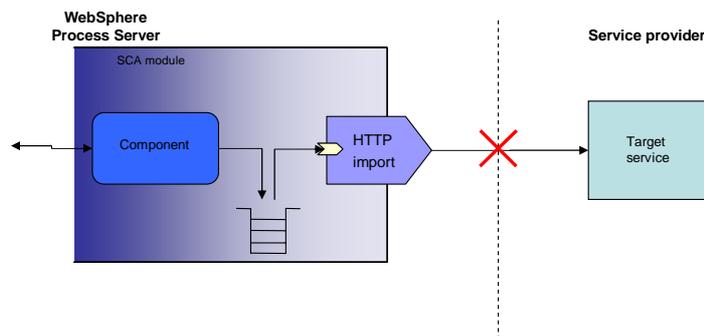
Store and forward feature

© 2010 IBM Corporation

This slide shows a scenario where you have an asynchronous import, which in this scenario is the JMS import shown in the picture. When the component makes a call to the import, the import interacts with the queue, which is the target service. In a scenario when the queue is unavailable, a `ServiceUnavailableException` is returned to the calling component. This exception is propagated back until it is handled or an asynchronous invocation is reached. A Failed event is generated at the first asynchronous invocation point. If the store and forward is not configured, any subsequent requests for the service will result in failed events being logged. If the store and forward is setup in this flow at the asynchronous invocation point, which in this case is at the JMS import, any subsequent requests for the target service will result in the messages being stored in a queue.

## Scenario 2- Synchronous imports – HTTP, EIS, EJB

- import unable to make a connection to the external service for example HTTP server



6

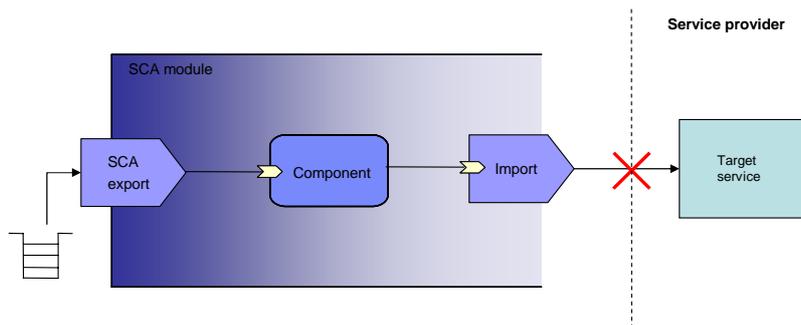
Store and forward feature

© 2010 IBM Corporation

This slide shows a scenario where you have a synchronous import - in this scenario, the HTTP import shown in the picture. Even though HTTP is a synchronous call, you can set the invocation style to asynchronous. When the component makes a call to the HTTP import, and the target service that the import interacts with is unavailable, a “service unavailable” exception is returned to the calling component. This exception is propagated back until it is handled or an asynchronous invocation is reached. A “Failed” event is generated at the first asynchronous invocation point. You can configure the store and forward in this flow at the asynchronous invocation point, which in this case is at the HTTP import. Any subsequent requests for the target service will result in the messages being stored in a queue.

### Scenario 3- Asynchronous SCA export

- All the invocations in this module are synchronous.
- SCA export is invoked asynchronously



7

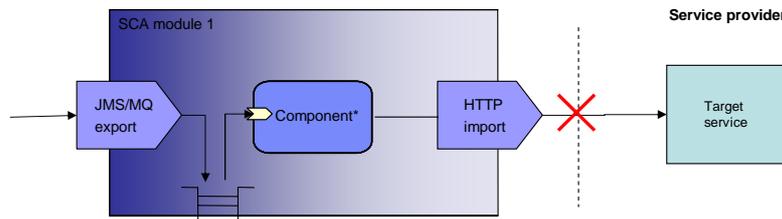
Store and forward feature

© 2010 IBM Corporation

The next supported case is for SCA export. In this scenario, you can see that all of the invocations are synchronous except for the invocation of SCA export. In this case, you can configure the store at the SCA export.

## Scenario 4- Asynchronous exports – JMS\*, MQ

- The JMS export always makes asynchronous invocations to the caller.
- The HTTP import is unable to communicate to its external service and hence returns an exception.



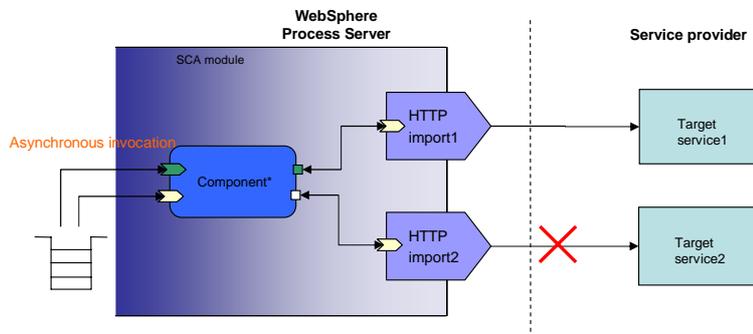
8

Store and forward feature

© 2010 IBM Corporation

The next scenario involves asynchronous exports like JMS and MQ. When the target service is unavailable in this scenario, an service unavailable exception is returned to the calling component. This exception is propagated back until it is handled or an asynchronous invocation is reached. In this scenario the JMS export always makes asynchronous invocations to the caller, so you can configure the store and forward qualifier at the component.

## Scenario 5 - Multiple interfaces and references

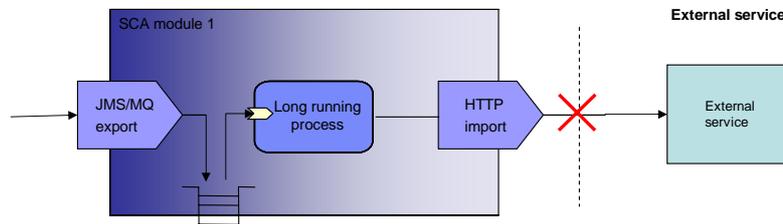


- Import2 is unable to make a connection to the target service
- The import returns ServiceUnavailableException to the calling component
- This exception propagates back until it is handled or an asynchronous invocation is reached
- A failed event is generated at the first asynchronous invocation point which is the component
- Ideally, HTTP Import1 should continue processing the events but this will depend on where the asynchronous invocation is

© 2010 IBM Corporation

The scenario here includes a component with multiple references and interfaces. In this scenario, when one of the targets (say for example the target service2) is unavailable, the asynchronous invocation point is at the component. So you can configure store and forward at the component. Any requests that come in to that component will now be stored. Notice that the target service 1 is still available - but even the request for the target service 1 will be stored. To avoid this, you can set the invocation style of the HTTP import1 and HTTP import2 as asynchronous and set the store and forward qualifiers at the import level. By doing this, only requests for a target service2 are stored and requests for target service1 will still go through.

## Scenario 6 - Long running business process – Not supported



- A long running process calls an external service synchronously
- The external service is unavailable hence the import returns an exception back to the long running process.
- Long running process will generate a failed event or a stopped activity
- Long running process does not support store and forward so the processing will continue and subsequent failed events or stopped activities are generated

This slide shows a scenario where you have a long running business process in your flow, which is the first asynchronous invocation point. Store and forward is not supported on long running business processes. Alternatively in this scenario, you can change the interaction style on the HTTP import to asynchronous and set the store and forward qualifier at the import level store the requests for the target service.

## Store and forward features

- Able to store the events proactively when the target service cannot be reached instead of creating failed events
- Resubmit queued events when the target service is available
- Specify the components where the store and forward should occur
- Widgets to monitor service control points at which the store is activated
  - Store or forward messages from widgets
- Can enable both event sequencing and store and forward

Store and forward provides you with the ability to proactively store events when the target service is unavailable instead of creating failed events. You can resubmit the events when the service is available. You can specify where you want the store to happen and what kind of exceptions trigger the store. Widgets are available in Business Space to monitor the state and either start the store or forward the messages from the store.

## How to configure store and forward

- determine where store and forward qualifier should be specified
  - like import of external service that might become unavailable
  - the component has to be invoked asynchronously
- exception specification can be specified at
  - component/export/import level
  - Individual interface level
  - Individual operation level

The first step before you can specify the store and forward qualifier is to identify the asynchronous invocation point in your flow. Store and forward can be used to store events for both synchronous and asynchronous services but the qualifier can only be set in your flow where there is an asynchronous invocation point. Once you identify the asynchronous invocation point, you can then decide what level to provide the qualifier at. You can configure the qualifier at the component, import, or export level, or at the individual interface level, or even at the operation level of an interface.

## Store and forward qualifier configuration in WebSphere Integration Developer (1 of 2)

The screenshot shows the WebSphere Integration Developer interface. In the background, there is an assembly diagram with a 'Stand-alone References' component. The 'Properties' panel for 'Component: Receiver (Java)' is open, with the 'Details' tab selected. The 'Details' tab shows a tree view with 'Component' selected. A yellow callout box points to this 'Component' level. In the foreground, the 'Add Qualifier' dialog box is open, showing a list of qualifiers: 'Data validation', 'Join activity session', 'Join transaction', 'Security permission', and 'Store and forward'. The 'Store and forward' qualifier is selected. Below the list, there is a checkbox for 'Determines if messages should be stored when an exception occurs.' and a text area with instructions: 'Store and forward qualifiers can be set on Interface Sets, Interfaces, and Interface Operations of SCA Components and Imports that can potentially be invoked asynchronously. It is recommended that you turn on "highlight synchronous calls" feature in order to assist you determine which interface sets are potentially invoked.' The dialog has 'Add', 'OK', 'Cancel', and 'Delete' buttons.

Select the level you want to set the qualifier, component level, interface level or operation level. In this example, you can see it is set at component level

13 Store and forward feature © 2010 IBM Corporation

You can see here the panel in WebSphere Integration Developer where you can set this qualifier. Once you identify the asynchronous invocation point, go to the properties view. In the properties panel, click the details tab. You can select the level at which you want to set the qualifier. Select the level and click Add to specify the store and forward qualifier.

## Store and forward qualifier configuration in WebSphere Integration Developer (2 of 2)

Specify if you want to look for a specific message to match in the exception chain and how the search should be done

Specify if you want to look for a specific message to match in the exception chain and how the search should be done

Exception that triggers the store. You can configure the type of exception by clicking on add button

14 Store and forward feature © 2010 IBM Corporation

Shown here is the screen capture of the assembly diagram and the properties panel in WebSphere Integration Developer. Once you add the qualifier, you can configure several attributes for it. You can configure what exception should trigger the store and if you want a particular message in the exception in addition to the exception itself. Provide a name for the qualifier configuration. This name is exposed to you in business space as service control point.

## Configuration – Exception specification

- Define qualifier policies
  - Exception name
    - Fully qualified exception name
    - Wild card character is \*
  - Exception message
    - Wild card character is \*
  - Exception chain
    - Look at the outermost exception in the exception chain
    - Look at all the exceptions in the exception chain starting from top
    - Look at all exceptions in the exception chain starting from bottom
    - Look at the innermost wrapped exception only
  - Exception hierarchy
    - Look for exceptions that subclass this exception

This slide shows some of the attributes that you can configure for a store and forward qualifier. You can configure what exception triggers the store and provide a message along with the exception as additional criterion. You can also specify how the message you provided should be matched in the exception, like “look in the outermost exception only” or “look at all the exceptions in the exception chain starting from the top.”

## Configuration – Exception specification

- Semantics of the specification
  - exception specification defined at component/import/export level
    - applies to all interfaces and operations on that component
  - exception specification defined at interface level
    - applies to that interface and all its operations. This specification applies in addition to the specification, if defined on the component of that interface.
  - exception specification defined at operation level
    - applies to that operation only. This specification applies in addition to the specification, if defined, on the component and the interface of that operation

Depending on what level you provide the exception, the conditions for the store to be triggered change. If you provide the exception at the component, import, or export level, the condition also applies to all the interfaces and individual operations for that component, export or import.

If you provide the exception at the interface level, the condition applies to that interface and all its operations in addition to the specification, if defined on the component of that interface

If you provide the exception at the operation level, the condition applies to that individual operation in addition to the specification, if defined, on the component and the interface of that operation.

## Clustered environment

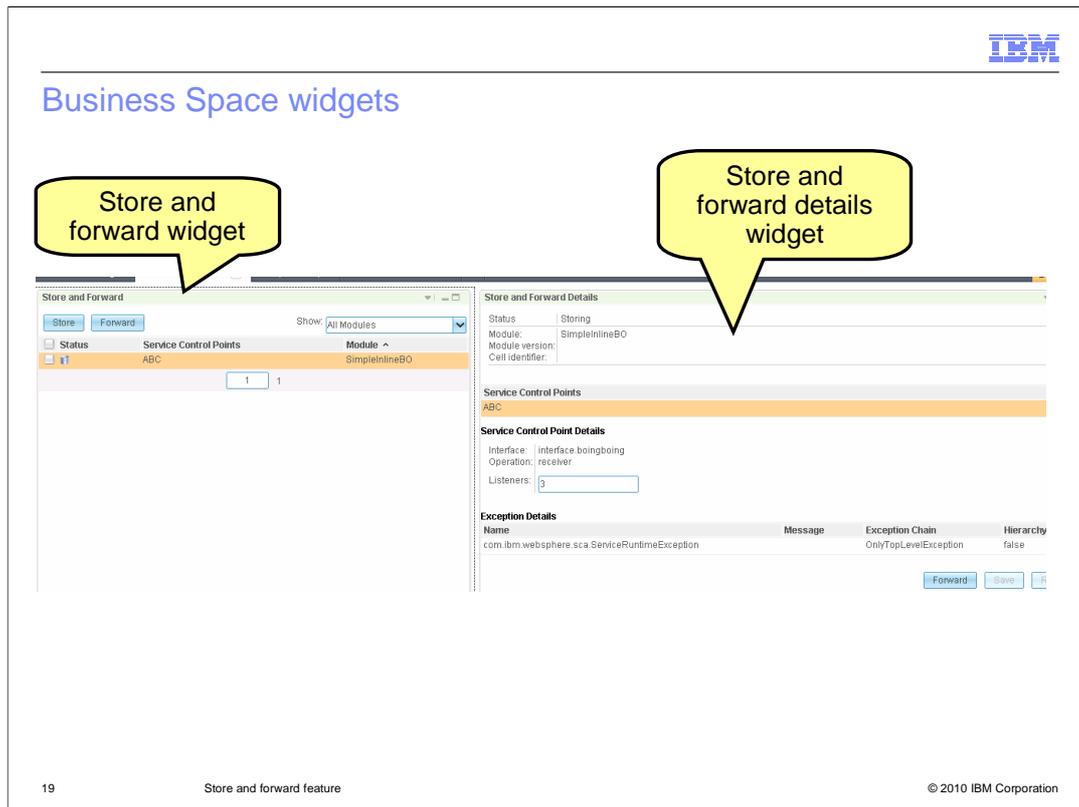
- If component in one of the cluster member, receives a failed request and starts the store, this state is propagated to the same component in all members of the cluster.
- Similarly, when one of the member receives a stop or start processing request from the user, this is propagated to all the servers in the cluster.
- There might be a timing window between the state changes on the different members due to propagation

The behavior of store and forward in a clustered environment is discussed in this slide. If the component, export, or import in one cluster member receives an exception that triggers a store at that member, the information to start storing is sent to the component, import, or export in rest of the cluster members. You can manually initiate or stop a store at a component, import, or export using business space. In this case, the same information is sent to the components in the rest of the cluster. There might be a timing between the state changes on different cluster members due to delay in the propagation. In that case, you can see more than one failed event generated and appear in the failed event manager.

## Store and forward in business space

- Store and forward qualifier configurations are exposed as service control points in business space
- Service control points can be monitored, controlled using these widgets
  - Module browser
  - Module administration
  - Store and forward
    - Lists all the service control points
  - Store and forward details
    - Lists the details of the service control point configuration

Business space provides a set of widgets for you to use to monitor and modify the state of the service control points. Service control points are the store and forward qualifier configuration you set up at component, import, or exports. Service control points are exposed in several widgets in business space such as module browser, module administration browser, specific widgets like the store and forward widget, and the store and forward details widget. The store and forward widget lists all of the service control points that are configured in the deployed applications. The store and forward details widget will provide more details on each selected individual control point in the store and forward widget, like the exception that triggers the store, present state of the store if it is storing or forwarding, number of listeners, and so on. You can modify the state of each control point from either the store and forward or the store and forward details widget.



On the left you can see the store and forward widget, which lists all of the service control points. In this scenario, there is only one control point. You can see the status of the control point, the name and the module in which it is configured. You can use the store and forward widget to initiate the store or stop the store by using the store and forward buttons. On the right, you can see the store and forward details widget, which provides details on the selected control point. You can see the exception that triggers the store, any message other than the exception that should be matched, and where to look for the message and the number of listener threads. You can also see the status of the service control point if it is storing or forwarding. You can use the store and forward details widget to initiate the store or stop the store by using the store and forward buttons.

## ***Summary***

This section will provide a summary of this presentation.

## Summary

- Store and forward
  - Scenarios supported and not supported
  - how to configure
  - properties that can be configured
  - Business space widgets

To summarize, when a target service is unavailable, an exception is returned. If store and forward qualifier is setup on this asynchronous component, then the store and forward service will compare the exception with the qualifier configuration. If the configuration matches, the store is activated on this component. Subsequent messages sent to this component are stored. Once the target service is available, the messages can be forwarded. The failed event that triggered the store has to be resubmitted separately. You also saw how to configure the store and forward qualifier, the various properties of the qualifier, and how it is exposed in the business space widgets.



## Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

[mailto:iea@us.ibm.com?subject=Feedback about WBPMv70 Store and Forward.ppt](mailto:iea@us.ibm.com?subject=Feedback%20about%20WBPMv70%20Store%20and%20Forward.ppt)

This module is also available in PDF format at: [../WBPMv70 Store and Forward.pdf](http://www.ibm.com/education/assistant/WBPMv70_Store_and_Forward.pdf)

You can help improve the quality of IBM Education Assistant content by providing feedback.



## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the Web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2010. All rights reserved.