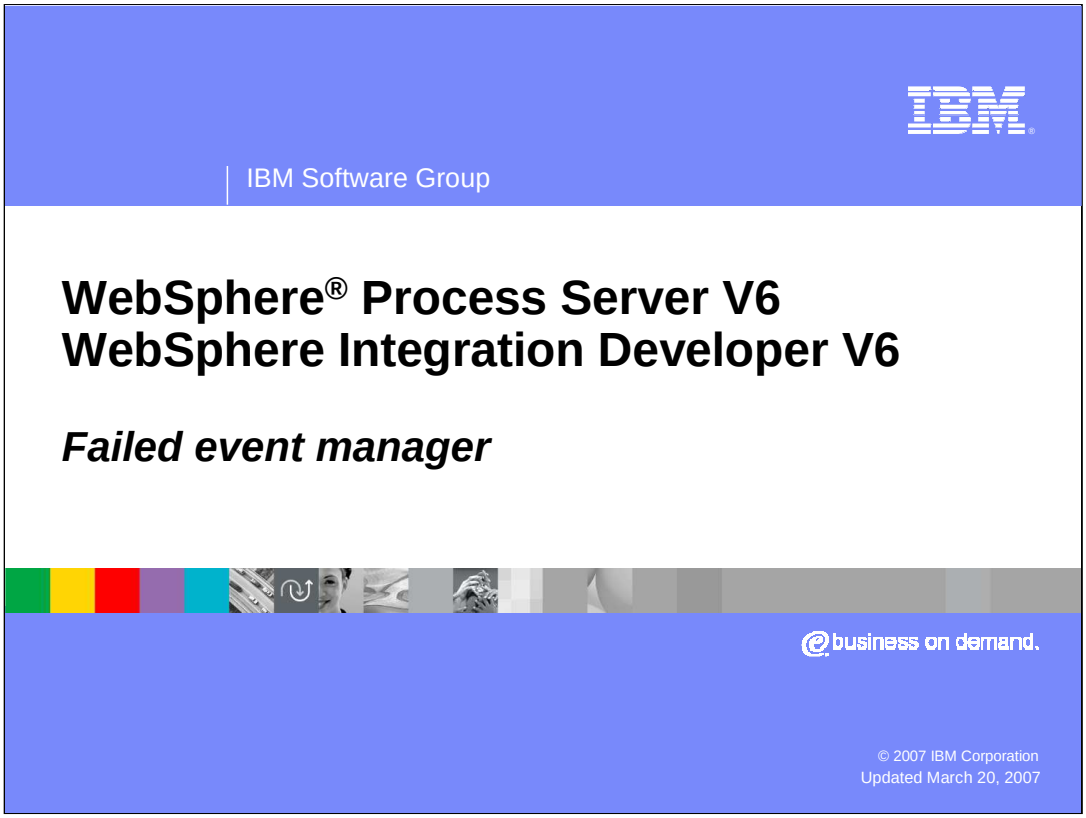


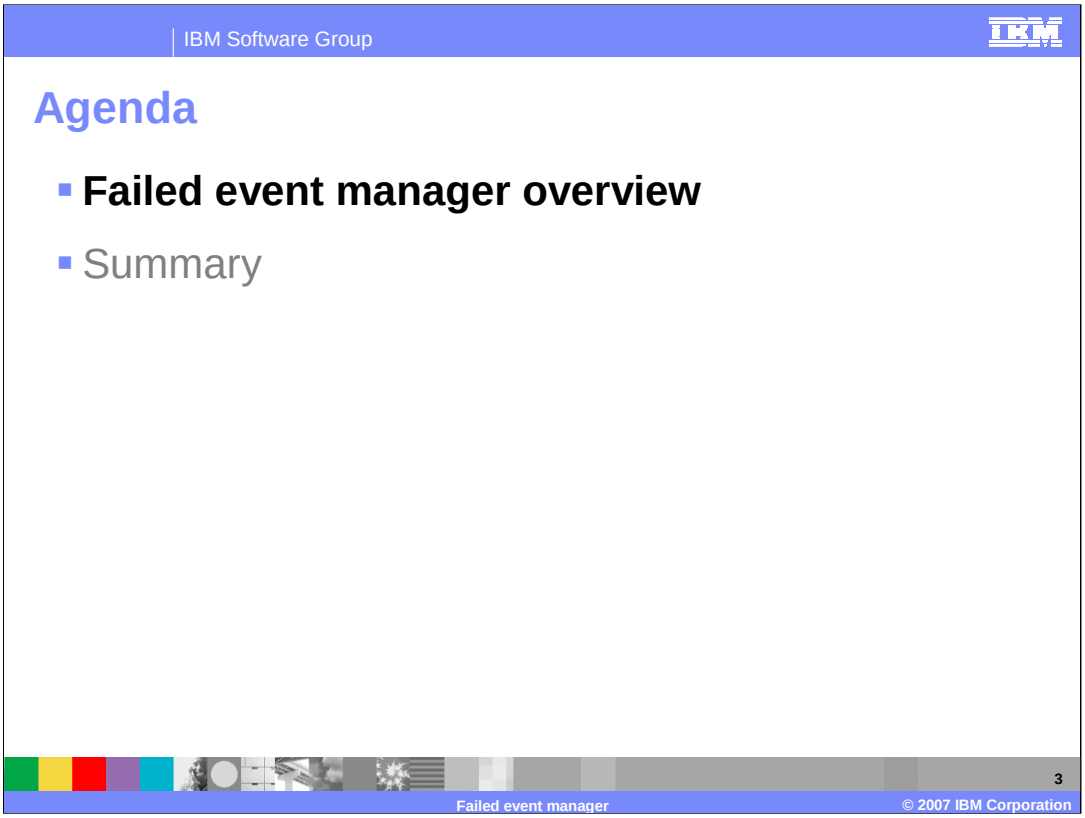
This presentation will focus on the failed event manager feature of WebSphere Process Server and WebSphere Integration Developer V6.0.

Goals

- Explain the failure recovery service included with WebSphere Process Server
- Describe the capabilities of the failed event manager

© 2007 IBM Corporation

The goals of this presentation are to cover the failure recovery service of WebSphere Process Server, and to describe the capabilities of the failed event manager.



This section will provide an overview of the failed event manager.

Failure recovery service for failed invocations

- WebSphere Process Server has a built in failure recovery service
- Captures and saves failed asynchronous one-way and two-way SCA invocations
  - Runtime exceptions only, business exceptions returned to caller
- Failed invocation information is persisted to data store
  - Source, destination, payload, Error (exception)
- Allows for resubmission of failed invocations
  - Manual resubmission of invocations

© 2007 IBM Corporation

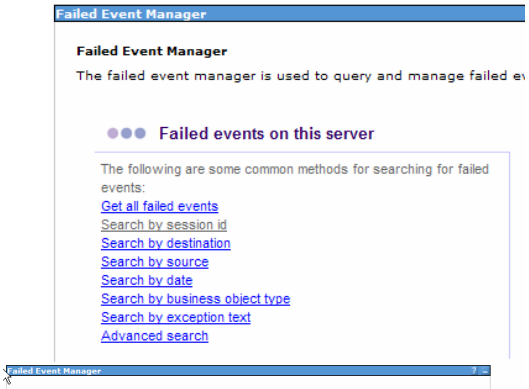
WebSphere Process Server includes a failure recovery service, which is a built-in service that monitors for failed operations between SCA components. For any asynchronous one-way or request/response SCA invocation that fails, after a number of retries, the failure recovery service will capture the request information and the failure and then store it. The type of failures which are caught are only from runtime exceptions. Any exceptions that are at the business level or defined as part of the component interface will not be caught. Business failures are returned to the component making the call. The information stored includes the source component, destination component, content being sent, and the error that occurred. Using this persisted data, the failed invocation can be then be resubmitted when the runtime problem has been resolved. This support helps reduce the problems in service-oriented environments where loose coupling can lead to service endpoints being unavailable or specified incorrectly.

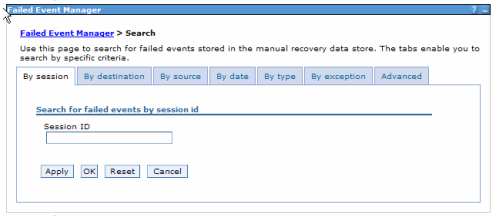
Failed events manager overview

- Web-based client for managing failed events included with WebSphere Process Server
  - Part of administrative console
- Query for failed events
  - By date, last successful event, last failed event, exception text, session ID

New V602

- Resubmit with detailed logging enabled
  - Single or multiple events
  - Change payload
- Delete
  - Single or multiple events

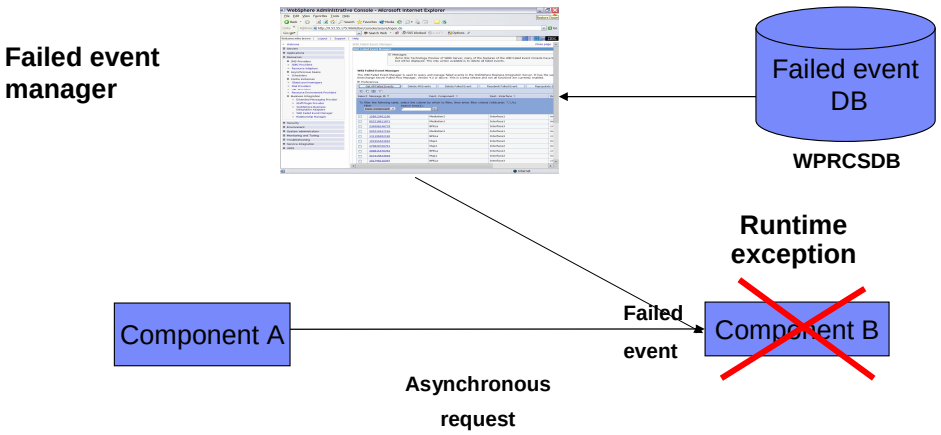




© 2007 IBM Corporation

WebSphere Process Server includes a Web-based client for working with and resubmitting the failed invocations. The failed event manager is an integration application and is available in the Administrative console. You can query for failed events using a variety of criteria such as date, last successful or failed event, by exception text, session ID or a combination of these. The session ID provides a consistent value that can be used to move to the Common Based Event Browser for the same failure. You can resubmit single or multiple failed events. While resubmitting, you can also change the payload. For instance, the failure could have been caused by passing in some inappropriate data. In this case, the payload can be updated from within the failed events manager and resubmitted. Only the data stored in memory would be updated, so the original source of the data will not be corrected. If a resubmitted event fails, this will show up as a new failed event in the failed event manager. There is also the ability to delete single or multiple events and this is often the appropriate action due to data becoming invalid since the time of the failure.

Failed event manager example



© 2007 IBM Corporation

To further emphasize the capabilities of WebSphere Process Server and the failed event manager, here is an example. Component A calls component B in an asynchronous manner. If Component B encounters a runtime exception, a failed event will be generated. The failure recovery service will capture this failure and store it in the failed event database. At this point, the system administrator opens the failed event manager to investigate the problem The administrator can query events, view failed activities details, re-submit events with logging enabled, or re-submit events with different level of logging and tracing. Once the runtime problem is resolved, the failed event can be resubmitted.

Agenda

- Administrative overview
- **Summary**

© 2007 IBM Corporation

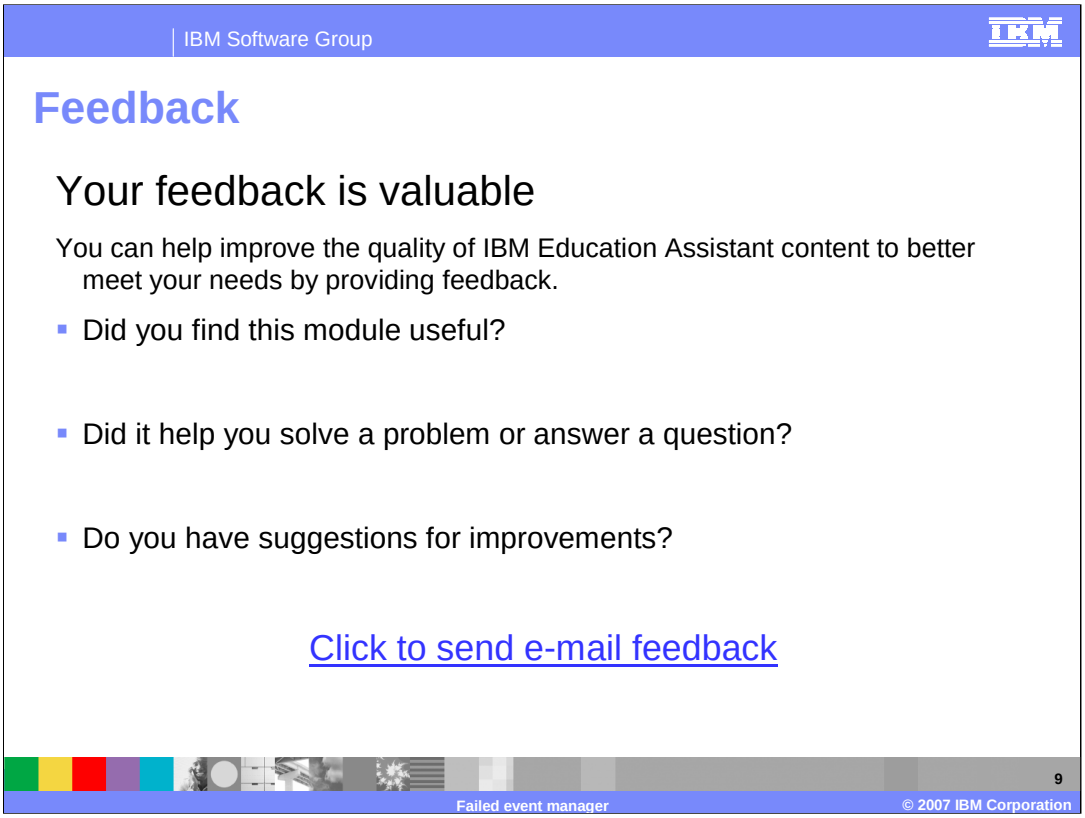
This section will provide a summary of the presentation.

Summary

- WebSphere Process Server provides a failure recovery service for capturing failed asynchronous requests resulting in runtime exceptions
- Web browser-based failed event manager available for investigating and resubmitting failed events

© 2007 IBM Corporation

The key points of this presentation are that WebSphere Process Server provides a robust failure recovery service for capturing failed asynchronous invocations that result in runtime exceptions. It also includes a Web browser-based application for working with these failed invocations. Through the failed event manager, failed requests can be resubmitted once the runtime problem has been resolved.



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

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM WebSphere

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2007. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

© 2007 IBM Corporation