Sterling B2B Integrator

FIFO Message Processing

Version 5.2

Sterling B2B Integrator

FIFO Message Processing

Version 5.2

Note

Before using this information and the product it supports, read the information in "Notices" on page 11.

Copyright

This edition applies to Version 5 Release 2 of Sterling B2B Integrator and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

FIFO Message Processing (V5.2.0 - 5.2.1)

Sterling B2B Integrator supports FIFO processing of messages through adapters.

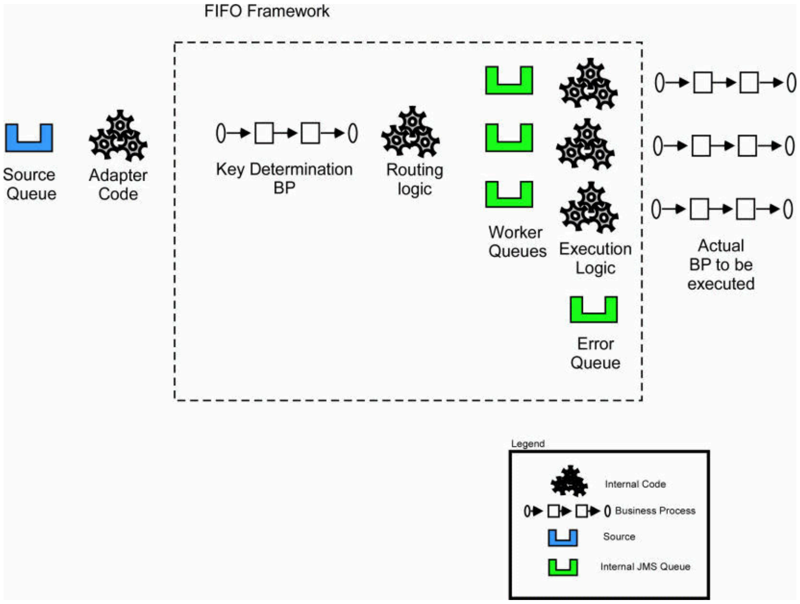
Introduction to FIFO Message Processing

Sterling B2B Integrator supports ordered processing of files and messages for the following adapters:

- JMS Queue adapter
- JMS Topic adapter
- MSMQ adapter

The ordered processing in Sterling B2B Integrator is processed by the FIFO (first in first out) framework.

The following figure demonstrates the FIFO framework:



Sterling B2B Integrator supports FIFO processing of messages through adapters. The messages passed to the FIFO framework are first executed through a specialized routing key initialization business process that returns a single string value known as the routing key. The routing logic is then applied, which places all the messages with equal keys on the same internal routing queue. Messages with different routing key values process in parallel. Messages with the same routing key value maintain FIFO ordering. Each queue to user specified business process processes the message and waits for the business process to end the metadata describing the errant process, then processes the next message. If an error is encountered while processing the messages, metadata describing the errant process are routed to an error queue. Thereafter, the message processing continues.

Configuring FIFO Execution

You can customize the name and number of queues used in the FIFO framework. The number of task queues determines the number of concurrent processes that can execute in the system at a time. You can increase the number of queues, but it will consume more resources.

The queue is defined in the `fifo.properties` property file in the `properties` directory. All settings in the `fifo.properties` configuration file can be overridden via `customer_overrides.properties`. See the `fifo.properties` file for additional information pertaining to customer overrides.

The default queue configuration is as follows:

```
workflow.taskqueue.2=FIFO.GIS.QUEUE.2
workflow.taskqueue.3=FIFO.GIS.QUEUE.3
workflow.taskqueue.4=FIFO.GIS.QUEUE.4
workflow.taskqueue.5=FIFO.GIS.QUEUE.5
workflow.taskqueue.6=FIFO.GIS.QUEUE.6
workflow.taskqueue.7=FIFO.GIS.QUEUE.7
workflow.taskqueue.8=FIFO.GIS.QUEUE.8
workflow.taskqueue.9=FIFO.GIS.QUEUE.9
workflow.taskqueue.10=FIFO.GIS.QUEUE.10
fifo.workflow.errorqueue=FIFO.GIS.ERROR
```

FIFO Error Elements

FifoError Element

The *FifoError Type* indicates the type of FIFO task that is being executed. At present, Async WorkFlow is the only type supported.

The table below lists the other FifoError elements:

Type	**Description**
TaskId	A unique ID given to each FIFO task executed by the FIFO framework.
TaskQueueId	The queue where the FIFO task was executed.
TaskQueueKey	The key that was returned through the FIFO routing key business process execution.
ErrorMessage	This element contains the information that assists in determining the cause of the failure.

WorkFlow Error Element

The table below lists the WorkFlow Error elements:

Type	**Description**
WorkFlowId	This element contains the workflow id that was executed.

Type	Description
WorkFlowContextId	This element contains the workflow context id for the first step of the business process. This information is used to retrieve the workflow and extract additional data in advanced scenarios.
WorkFlowInitiator	This element contains the name of the workflow initiator. In most cases, name of the adapter that started the process will be the workflow initiator name.
PrimaryDocumentId	This element contains the ID for the primary document of the business process.

FifoInitializationBPReport

This element contains metadata that describes the execution of the routing key initialization business process.

This is an optional node. It will be included both in process data of the executed business process and in the error queue XML. It is automatically included in the XML data if an error occurs during task initialization. To force the inclusion of this data, both in the error report and process data of the executed business process, ForceFifoInitializationDump to "true" in the routing key business process

The table below lists the initialization BP report elements:

Type	Description
AdvancedStatus	This element contains the advanced status for the final step of this business process.
BasicStatus	This element contains the basic status for the final step of this business process.
PrimaryDocumentId	This element contains the primary document id at the last step of this business process.
ServiceName	This element contains the service name for the last step of this business process.
wfdName	This element contains the workflow definition name for this business process.
wfdVersion	This element contains the workflow definition version for this business process.
WorkFlowContextId	This element contains the workflow context id for this business process.
WorkFlowID	This element contains the workflow id for this business process.
StatusReport	This element contains the status report, if any, at the last step of this business process.
ProcessData	This element contains the process data at the last step of the business process.

FifoErrorNode Element

When the routing key business process is executed, the business process author can optionally write additional metadata to the FifoErrorNode element in the process data. This element and all the child nodes will be included in the FifoError document as part of this element.

The routing key business process has access to all process data information passed onto it through the adapter. See the example below for additional information about generating an error node.

```
<process name="AssignQueueKey">
  <sequence>
 <assign to="FifoRoutingKey"    from="DocToDOM(PrimaryDocument)/Order/@OrderId" />
   <assign to="FifoErrorNode/MSMQ/@QueueName" from="string(MSMQ/@QueueName)"  append="true"/>
  </sequence>
</process>
```

The additional information from the adapter can be included in the element to preserve the context of the error information in an easily identifiable manner.

FIFO Error Queue Listener

An out of the box adapter is configured on each node to listen to the error queue. This adapter is named "FIFO Error Queue Listener {nodename}." The adapter will bootstrap a business process named FifoError. This process is configured to retrieve the data from the errant process, including the original document and to integrate it into this process. This allows you to automate the re-processing of the data and other activities.

The FifoError process is defined as follows:

```
<process name="FifoError">
 <sequence>
    <operation>
     <participant name="FIFORouting" />
       <output message="Xout">
          <assign to="." from="*"></assign>
         <assign to="FifoTask">FifoErrorRecord</assign>
       </output>
        <input message="Xin">
         <assign to="." from="*"></assign>
       </input>
    </operation>
 </sequence>
</process>
```

The FifoError process provides a basic implementation for error handing. A user-specified business process may be configured to allow for customized error handling. A user-specified business process must contain the FIFORouting service as configured in the default FifoError process.

Recovering Errant Data

The messages in the error queue are written in XML format. The XML format provides information to determine the nature and source of the document containing the error. The error message contains information that enables the retrieval of document data; however, contents of the document are not stored in the message. The error message format is as below:

```
<?xml version="1.0" encoding="UTF-8"?>
<FifoError ErrorMessage="" ErrorType="" TaskId="" TaskQueueId="" TaskQueueKey="" Type="">
  <WorkFlowError PrimaryDocumentId="" WorkFlowContextId="" WorkFlowId="" WorkFlowInitiator="">
   <FifoErrorNode/>
    <FifoInitializationBpReport AdvancedStatus="" BasicStatus="" PrimaryDocumentId="" ServiceName=
      WorkFlowContextId="" WorkFlowId="">
        <StatusReport></StatusReport>
       <ProcessData>
          <PrimaryDocument SCIObjectID=""/>
        </ProcessData>
   </FifoInitializationBpReport>
  </WorkFlowError>
</FifoError>
```

FIFORouting Service

The FIFORouting service provides a control and reporting mechanism for interaction between business processes and the FIFO subsystem.

The FifoTask parameter specifies the task that this service should execute. Currently, there are two operational tasks this service provides: FifoResponse and FifoErrorRecord.

The FifoErrorRecord parameter specifies that the FIFORouting service should parse an error record from the error queue, retrieve the errant business process data, and report on it, as described above. This parameter should be used in conjunction with a retrieval of an error record from the error queue. The primary document in this mode of operation must be an FifoError XML record.

When executed in the FifoErrorRecord mode, the FIFORouting service will retrieve data pertaining to the errant business process and include it in ProcessData for the current business process. All data, including documents, may then be used directly within the current business process. The service will generate data of the following format:

```
<ProcessData>
 ...
 <PrimaryDocument SCIObjectID=""/>
...
 <FifoProcess ErrorType="" WorkFlowContextId="" WorkFlowId=""
     WorkFlowInitiator="">
    <ProcessData>
     <FifoDetails>
        <FifoInitializationBpReport AdvancedStatus="" BasicStatus=""
           PrimaryDocumentId="" ServiceName="" WfdName="" WfdVersion=""
           WorkFlowContextId="" WorkFlowId="">
         <StatusReport>
          </StatusReport>
         <ProcessData>
            <PrimaryDocument SCIObjectID="" />
          </ProcessData>
       </FifoInitializationBpReport>
    </FifoDetails>
    </ProcessData>
 </FifoProcess>
</ProcessData>
```

Note: The first instance of ProcessData is that of the current error handler business process. The FifoProcess element contains the data from the errant business process. The ProcessData element within this element contains the data from the original errant business process. All data and documents within this ProcessData element may be used directly within this business process for error handing purposes.

The FifoReponse parameter specifies that the FIFORouting service should return a positive or negative success response to the FIFO subsystem. An optional parameter, FifoStatus, may also be specified. This status indicates whether or not the business process was a success and if it is an error, designations the FIFO subsystem to report an error. The FifoStatus parameter considers ERROR to be a failure and any other string data to be success.

The FifoResponse parameter is used to provide early response at to the success or failure of a FIFO business process. For example, assume business process A is the process that must be executed in FIFO. Business process A contains 10 steps. The first five steps must be executed in order; however, the last five steps provide data execution functionality where order is not important. In this example, optimal performance will be achieved by utilizing the FIFORouting service in FifoResponse mode to return the response at step 6. This will allow the next message to be processed immediately following the execution of this service and allow steps 7 through 11 to execute fully parallel.

Business Process Error Queue

The business process error queue is defined within the `fifo.properties` file. The error queue configuration defines the destination of errors within the FIFO framework. The error queue name should not contain spaces or punctuation.

The default business process error queue is:

```
fifo.workflow.errorqueue=FIFO.GIS.ERROR
```

Business Process Queues

The FIFO business process execution queues are defined by rows that are prefixed with workflow.taskqueue. A queue row consists of a unique ID with prefix workflow.taskqueue to the left and a unique name without spaces or punctuation to the right.

You can add a queue by adding an additional row to the existing property file or to `customer_overrides.properties`. The simplest way to add additional queues is to continue the existing numbering scheme. You can remove a queue by deleting a row.

Note: Queues cannot be reduced below their default set of ten queues using `customer_overrides.properties`. If this is required, the queues must be removed directly from `fifo.properties`.

FIFO processing must be complete and the queues must be empty to change the queue configuration. You must disable the inbound adapter while changing the queue configuration. If the inbound adapter is not disabled and the queues are not drained, it may result in message execution that is out of order.

Cluster Configuration

About this task

The FIFO messaging system requires an external clustered JMS provider to allow proper execution and failover in a clustered configuration. An out of the box configuration for ActiveMQ 5.2 is provided to streamline this deployment. To configure FIFO messaging in a cluster for ActiveMQ:

Procedure

1. Download ActiveMQ 5.2 from http://activemq.apache.org/activemq-520-release.html for the appropriate OS.
2. Deploy an instance of ActiveMQ 5.2 on each node of the cluster.
3. An activemq.xml file is included the properties/fifo directory of the Sterling B2B Integrator deployment of each node. For each node, take this file and copy it to the ActiveMQ deployment on that node within the "conf" directory. This file will configure ActiveMQ to use failover clustering utilizing the Sterling B2B Integrator database for storage and configure its port usage. By default, ActiveMQ will be configured to listen at the Sterling B2B Integrator base port + 65 and the ActiveMQ interface will be at base port + 66 (http://server:base port + 66/admin).
4. On each Sterling B2B Integrator node, the queue configuration must be re-directed to utilize the ActiveMQ cluster. In each node, add the following to `customer_overrides.properties`:
   ```
   fifo.broker.username=fifo.broker.password=fifo.broker.url=failover:(tcp://node1_hostr
   + 65,tcp://node2_hostname:node_2_base_port + 65, ...,
   tcp://noden_hostname:node_n_base_port + 65)
   ```
5. Start the ActiveMQ instances on each node. See http://activemq.org for additional information about running an ActiveMQ instance.
6. Restart Sterling B2B Integrator.

Configure FIFO Services

About this task

To configure FIFO services:

Procedure

1. Login to Sterling B2B Integrator.
2. Select **Deployment** > **Services** > **Configuration**.

3. Create new service and click **Go!**
4. In the Service Type field, enter the applicable adapter you want to use and click **Next**. You can also select it from the Tree View or List View.
5. Enter a suitable name and description in **Name** and **Description** fields.
6. Select or create a new group if required. By default, it is None.
7. Select the business process you want to execute.

   Note: This business process must be set to use at least Minimal Event Processing and cannot be set to Error Only persistence level.
8. Select **FIFO** from Processing Mode drop-down list and click **Next**.
9. Select the business process that will receive the message and returns the routing key from the **FIFO Route Lookup BP** drop-down list. You should create a business process and import it into Sterling B2B Integrator.
10. Review and click **Finish**. The service is saved and the system displays a `The system update completed successfully` message.

Example

The example below demonstrates routing key business process, which executes a set of XML documents in FIFO order by OrderID field:

```
<process name="AssignQueueKey">
  <sequence>
   <assign to="FifoRoutingKey"    from="DocToDOM(PrimaryDocument)/Order/@OrderId" />
  </sequence>
</process>
```

The routing information is not limited to XML documents only. Translation, Document Extraction, and other data extraction services can also be employed to retrieve routing data. In addition to the routing information in the document, the routing key business process has access to all information passed from the adapter in process data. If the routing key process fails, the error information will be placed in the *Business Process Error Queue*.

The routing key process must be configured with the *Enable Async Start Mode* disabled via the routing business process manager. If this is not configured, the routing key process will fail and the error information will be placed in the error queue.

Note: The FIFO Routing adapter must be enabled for message processing to occur. If this adapter is not enabled, messages will remain on the internal FIFO routing queues and no processing will occur.

ActiveMQ Data Storage

JDBC (Database) Master Slave is the default configuration for data storage employed to store FIFO data for ActiveMQ. In this configuration, each ActiveMQ node in a cluster is configured to utilize a single, shared database.

By default, this option is configured to make use of the existing Sterling B2B Integrator database. As a result, this configuration option is setup out of the box and provides the simplest storage solution.

Shared File System Master Slave is an alternative data storage mechanism supported for FIFO, where a shared file system is used to store FIFO data for ActiveMQ. The shared file system option may yield better performance than when using JDBC.

Configure Shared File System Master Slave for ActiveMQ

About this task

You must manually configure the Shared File System Master Slave if you are not using the JDBC (Database) Master Slave configuration option to store FIFO data for ActiveMQ.

Note: Configuring FIFO messaging in a cluster for ActiveMQ is a prerequisite to configure the Shared File System Master Slave for ActiveMQ. For information on configuring FIFO messaging in a cluster, see *Cluster Configuration*.

To configure shared file system master slave for ActiveMQ:

Procedure

1. In the activemq.xml file, comment out the following section:

   XML comments consist of the symbols, '<!--' to open the comment and '-->' to close the comment.

   ```
   <!-- Database Storage Option -->

   <!-- This section has been commented.

   <persistenceAdapter>
         <jdbcPersistenceAdapter dataSource="#fifo-ds" useDatabaseLock="true">
           <statements>
             <statements tablePrefix="FIFO_"/>
           </statements>
         </jdbcPersistenceAdapter>
   </persistenceAdapter>

   -->
   ```
2. Uncomment the following section by removing the symbols '<!--' and '-->'.

   ```
   <!-- File system Storage Option -->

    <persistenceAdapter>
       <journaledJDBC dataDirectory="/sharedFileSystem/broker"/>
   </persistenceAdapter>
   ```
3. Edit the dataDirectory parameter to point to the location of the shared data directory to be used. This data directory must point to the same physical data location for all ActiveMQ instances in the network. For information on warnings about shared file system choices as a result of locking limitations, see *Shared File System Assumptions and Limitations*.
4. Restart each ActiveMQ node when you reconfigure it.

Shared File System Assumptions and Limitations

The following are some of the assumptions and limitations you must be aware of when using the Shared File System option to store FIFO data for ActiveMQ.

- Encrypted passwords for database storage are not currently supported. The file system based storage option described in this topic provides an alternative that does not require you to expose the database passwords.
- If an ActiveMQ node loses its connection to its database or file system storage, ActiveMQ will shut down. This is the intended behavior. Sterling B2B Integrator

currently does not employ out of the box monitoring for the ActiveMQ instances utilized for FIFO. To ensure seamless FIFO processing, the ActiveMQ nodes must be monitored and restarted if the instances are shut down for any reason.

- When ActiveMQ loses its database connection in conjunction with a Microsoft SQLServer database, ActiveMQ may hang during the shut down process. As a result, it may be difficult to determine if the ActiveMQ node has failed and requires to be restarted. Should you use the shared file system storage when using ActiveMQ in combination with a SQL Server database to avoid processing interruptions in failure scenarios.
- If you are reconfiguring any ActiveMQ options, ensure that you have executed all FIFO business processes. Failure to execute all FIFO business processes may result in the existing FIFO business processes remaining in an ‘Active' state, in turn resulting in loss of FIFO ordering for the processes in the ‘Active' state. To continue successful processing, the business processes in the ‘Active' state will have to be manually halted and restarted.

Notices

This information was developed for products and services offered in the U.S.A.

IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA 95141-1003

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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 and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2015. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2015.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® 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 IBM trademarks is available on the Web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center®, Connect:Direct®, Connect:Enterprise®, Gentran®, Gentran®:Basic®, Gentran:Control®, Gentran:Director®, Gentran:Plus®, Gentran:Realtime®, Gentran:Server®, Gentran:Viewpoint®, Sterling Commerce™, Sterling Information Broker®, and Sterling Integrator® are trademarks or registered trademarks of Sterling Commerce®, Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.

Product Number:

Printed in USA