

webMethods Audit Logging Guide

Version 8.0

December 2009

This document applies to webMethods Product Suite Version 8.0 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2005-2009 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

The name Software AG, webMethods, and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products." This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

Table of Contents

About This Guide	5
Document Conventions	5
Documentation Installation	6
Online Information	6
1. Concepts	9
Overview	10
Error Audit Logging	11
Session Audit Logging	11
Service Audit Logging	11
Security Audit Logging	12
Document Audit Logging	13
Guaranteed Delivery Audit Logging	13
Business Process Audit Logging	13
Task Audit Logging	14
Integration Process Audit Logging	14
Mediator Transaction Logging	15
Globalization	15
2. Setting Up Audit Logging	17
Overview	18
Configure Audit Logging	18
Start the Logger Configuration	18
Enable the Logger	18
Choose the Logging Level for the Service Logger	19
Choose the Mode for Writing Log Entries	19
Indicate Whether to Persist the Queue	20
Specify the Maximum Queue Size	20
Identify the Destination	21
Specify the Maximum Retries for Database Destination	22
Specify the Wait Between Retries for Database Destination	22
Additional Fields for the Security Logger	23
Choose Whether to Generate Auditing Data on Startup	23
Choose When to Generate Security Auditing Data	23
Choose Security Areas to Audit	23
Complete the Logger Configuration	23
Set Up Additional Service Logging	24
Set Up Customized Service Logging in Developer or Designer	24
Write User-Defined Messages or Input Pipelines to the Integration Server Server Log	25
Write User-Defined Messages to the IS Core Audit Log	25
Send Messages About Service Failures to Email Addresses	26
Perform Additional Processing on Audit Log Entries	27

3. Viewing Audit Log Data	29
Overview	30
View the Audit Logs in Integration Server Administrator	31
View the Error Log	31
View the Guaranteed Delivery Log	31
View the Security Log	32
View the Service Log	32
View the Session Log	33
View the Mediator Transaction Logs	34
Change the Log Displays	36
Display Logged Data in Different Languages	36
Change the Display Permanently for All Logs	37
Change the Display Temporarily for a Particular Log	38

About This Guide

This guide explains how to configure webMethods Integration Server error, session, service, security, document, and guaranteed delivery audit logging, and how to view logged data. In addition, the guide briefly describes business process, task, integration process audit, and webMethods Mediator transaction event logging, and points to the webMethods documentation that provides more detailed information for those types of logging.

Document Conventions

Convention	Description
Bold	Identifies elements on a user interface.
Narrow font	Identifies storage locations for services on webMethods Integration Server, using the convention <i>folder.subfolder.service</i> .
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
<i>Italic</i>	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
Monospace font	Identifies text you must type or messages displayed by the system.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Documentation Installation

You can download the product documentation using the Software AG Installer. Depending on the release of the webMethods product suite, the location of the downloaded documentation will be as shown in the table below.

For webMethods...	The documentation is downloaded to...
6.x	The installation directory of each product.
7.x	A central directory named <code>_documentation</code> in the main installation directory (webMethods by default).
8.x	A central directory named <code>_documentation</code> in the main installation directory (Software AG by default).

Online Information

You can find additional information about Software AG products at the locations listed below.

 **Note:** The Empower Product Support Web site and the Software AG Documentation Web site replace Software AG ServLine24 and webMethods Advantage.

If you want to...	Go to...
Access the latest version of product documentation.	Software AG Documentation Web site http://documentation.softwareag.com
Find information about product releases and tools that you can use to resolve problems. See the Knowledge Center to:	Empower Product Support Web site https://empower.softwareag.com
<ul style="list-style-type: none">■ Read technical articles and papers.■ Download fixes and service packs.■ Learn about critical alerts.	
See the Products area to:	
<ul style="list-style-type: none">■ Download products.■ Get information about product availability.■ Access older versions of product documentation.■ Submit feature/enhancement requests.	

If you want to...	Go to...
<ul style="list-style-type: none">■ Access additional articles, demos, and tutorials.■ Obtain technical information, useful resources, and online discussion forums, moderated by Software AG professionals, to help you do more with Software AG technology.■ Use the online discussion forums to exchange best practices and chat with other experts.■ Expand your knowledge about product documentation, code samples, articles, online seminars, and tutorials.■ Link to external Web sites that discuss open standards and many Web technology topics.■ See how other customers are streamlining their operations with technology from Software AG.	<p>Software AG Developer Community for webMethods</p> <p>http://communities.softwareag.com/webmethods</p>

1 Concepts


■ Overview	10
■ Error Audit Logging	11
■ Session Audit Logging	11
■ Service Audit Logging	11
■ Security Audit Logging	12
■ Document Audit Logging	13
■ Guaranteed Delivery Audit Logging	13
■ Business Process Audit Logging	13
■ Task Audit Logging	14
■ Integration Process Audit Logging	14
■ Mediator Transaction Logging	15
■ Globalization	15

Overview

Audit logging for webMethods products provides important data you need to monitor webMethods system activity and correct problems. Integration Server maintains most of the audit logging data in the webMethods product suite. This guide describes audit logging maintained by Integration Server, as follows:

This type of logging...	Provides information about...
Error	Exceptions thrown by services running on Integration Server.
Session	Sessions opened on Integration Server by webMethods Developer, webMethods Designer, third-party clients, and other servers.
Service	Services that run in Integration Server.
Security	Security-related administrative and operational actions on Integration Server, such as modifications to authorizations and authentication, and attempts to access Integration Server resources or perform runtime events.
Document	Documents that are in doubt, have failed, or have exhausted trigger retries on Integration Server.
Guaranteed delivery	Guaranteed delivery transactions in Integration Server.
Business process	Business processes modeled in Designer that run on Integration Servers.
Task	Tasks designed in Designer that run on My webMethods Server. Tasks can be called from business processes or can run as standalone tasks.
Integration process	Integration processes made up of a chain of services that run on Integration Servers.
Mediator transaction	webMethods Mediator transaction events produced by the Log Invocation run-time policy.

This chapter describes each type of audit logging. In addition, the chapter lists the default language used for log entries and describes the effect of your operating environment and webMethods language packs on log entries.

 **Note:** For information on setting up audit logging for webMethods adapters, see the adapter guides.

Error Audit Logging

Error audit logging provides data about exceptions thrown by services running on Integration Server. You can use error log data to debug services. Sample stack trace information is shown below.

```
java.lang.NullPointerException
at JpLogger.addScheduleID(JpLogger.java:46)
at java.lang.reflect.Method.invoke(Native Method)
at com.wm.app.b2b.server.JavaService.baseInvoke(JavaService.java:287)
at com.wm.app.b2b.server.ServiceManager.invoke(ServiceManager.java:344)
at com.wm.app.b2b.server.comm.DefaultServerRequestHandler.handleMessage
DefaultServerRequestHandler.java:97)
at com.wm.app.b2b.server.HTTPMessageHandler.process(HTTPMessageHandler.java:167)
at com.wm.app.b2b.server.Dispatch.run(Dispatch.java:204)
at com.wm.util.pool.PooledThread.run(PooledThread.java:105)
at java.lang.Thread.run(Thread.java:498)
```

Session Audit Logging

Session audit logging provides data about sessions opened on Integration Server by Developer, Designer, third-party clients, and other servers.

You can use session log entries to do the following:

- Track when sessions start, their current status, and their duration.
- Record the client that initiated the session and the Integration Server port on which the client connected.

Service Audit Logging


Service audit logging provides data about flow and coded (for example, Java) services that run in Integration Server. You can use service log entries and data to do the following:

- Track when services start, their status, and their duration.
- Track whether services completed successfully or failed.
- Record the client that called the service, and the Integration Server port on which the client connected.
- Resubmit services.

In Integration Server, you globally disable all logging for all services, globally enable one type of logging for all services, or enable customized logging on a service-by-service basis.

If you enable customized logging, you set up the customized logging for specific services in Developer or Designer. For each service, you can choose the following:

- When to log based on how the service is called. For example, you might choose to log only when the service is called by a client request or trigger, as opposed to by other services.
- On what status to log. For example, you might choose to log only when the service fails.
- Whether to store the service's input pipeline and, if so, when. For example, you might choose to log the input pipeline only when an error occurs. Storing the input pipeline allows you to resubmit the service later if necessary.

 **Note:** Whether you enable or disable service logging in Integration Server and Developer or Designer, if error logging is enabled, Integration Server always writes error log entries when service errors occur. The data includes stack trace data about the errors.

You can augment service logging data using Integration Server built-in services. The built-in services do the following:

- Enable services to post user-defined progress messages to the Integration Server server log or the IS Core Audit Log. For example, you might have a service post messages to indicate that certain pieces of code ran successfully, or to record run-time values for variables so you can see how the values changed as the service ran.
- Enable services to write the pipeline to the Integration Server server log.

Security Audit Logging

Security audit logging provides data about security-related administrative and operational events that occur on Integration Server. Administrative events are configuration changes related to Integration Server security activities. Examples include changes to authorization, authentication, port, or audit settings; SSL configuration, password restrictions; or root certificates. Operational events include attempts to log on to Integration Server and to access Integration Server services and documents.

You can use security log entries to do the following:

- Track security events that occurred, when they occurred, and by whom they were performed; includes log entries about enabling or disabling security auditing in general and for particular areas (for example, authentication).
- Track whether events completed successfully or failed.

Document Audit Logging

When a trigger is configured for exactly-once processing and Integration Server cannot determine whether the current document is a copy of one the trigger has already processed, Integration Server logs the document to the external RDBMS as an *in doubt* document.

If a transient error occurs while Integration Server is publishing, delivering, or retrieving a document for a trigger, Integration Server logs the document to the external RDBMS as a *failed* document.

If Integration Server has tried repeatedly to publish or deliver a document for a trigger from its outbound store and failed, Integration Server logs the document to the external RDBMS as a *retries exceeded* document.

For complete information, see the *Publish-Subscribe Developer's Guide*.

Guaranteed Delivery Audit Logging

If you configure the guaranteed delivery capability in Integration Server, guaranteed delivery audit logging provides data about guaranteed delivery transactions. You can use guaranteed delivery log entries to do the following:

- Track when transactions start and their current status.
- See the names of guaranteed delivery processes that are running.
- Track whether the processes completed successfully or failed.

For complete information about Integration Server's guaranteed delivery capability, see *Administering webMethods Integration Server*.

Business Process Audit Logging

Business process audit logging provides data for business processes modeled in Designer that run on Integration Servers. You can use business process log entries and data to do the following:

- Identify business processes.
- Record the path that business processes took at run time.
- Track when business processes and business process steps started, changed status, and ended.
- Track whether business processes and steps completed successfully or failed.
- Resubmit business processes at specified steps.

In Designer and the Monitor user interface, you specify the amount and type of data to log for each business process model version. In Designer, you can also specify process step input and output document fields for which to log run-time values. In the Monitor user interface, you can also choose to log process transitions so you can see the path the process took at runtime. For instructions on setting up business process logging, see the Designer online help and *Monitoring BPM, Services, and Documents with BAM: webMethods Monitor User's Guide*.

Process Engines log audit data for business processes. The Process Engine is a package installed on every Integration Server that runs business process steps. For detailed information on the Process Engine and how it logs data, see *Administering webMethods Process Engine*.

Task Audit Logging

Tasks are created in Designer and run on My webMethods Server. You can log two types of audit data for tasks:

- For all tasks, you can use task log entries to track the following:
 - When tasks are queued.
 - When users accept or release tasks, suspend and resume tasks, and complete or cancel tasks.
 - Whether tasks completed successfully, failed, or expired.

Task Engines log audit data for tasks and send the data to Integration Server. The Task Engine is a feature installed on every My webMethods Server that runs tasks. For detailed information on the Task Engine and instructions on setting up task logging, see *Working with BPM Tasks: webMethods Task Engine User's Guide*.

Users perform the actions listed above from the task list in My webMethods. For instructions on performing actions on tasks, see *Working with BPM Tasks: webMethods Task Engine User's Guide*.

- For tasks that are called from business processes, you can write business process log entries. Tasks called from business processes are run as business process steps, so you can log the same data for a task that you can log for any other business step (see [“Business Process Audit Logging”](#), above). Process Engines log all business process entries.

Integration Process Audit Logging

You can log entries that track the progress and results of integration processes. To do so, you have the services that make up the integration process call webMethods Monitor built-in services that create these entries. For complete information, see *Monitoring BPM, Services, and Documents with BAM: webMethods Monitor User's Guide*.

Mediator Transaction Logging

You can log Mediator transaction events produced by the Log Invocation run-time policy. This policy is enforced in Mediator, but you identify the audit log as the destination for the events in CentraSite. You can use Mediator transaction log entries to do the following:

- Identify the SOAP session, virtual service, and instance of Mediator on which the transaction events occurred.
- If the Identify Consumer policy action is defined for the virtual service, identify the IP address and name of the service consumer.
- Track whether events completed successfully or failed.
- Record the content of request and response payloads for service calls.


 **Note:** You can only log request and response payloads if you are writing Mediator transaction events to an external RDBMS.

For information about identifying the audit log as the destination, see the CentraSite ActiveSOA documentation. For information about enforcing policies with Mediator, see *Administering webMethods Mediator*.

Globalization

If a webMethods product is equipped with webMethods language packs and some of those language packs correspond to the language used by the operating environment in which the product is running, the product writes its log entries in the language used by the operating system. If the product is equipped with no language packs or with language packs that do *not* correspond to the language used by the operating system, the product writes its log entries in U.S. English.

Suppose your operating environment uses Japanese as its language. You have installed language packs including the Japanese Language Packs on Integration Server, so Integration Server stores its own log entries in Japanese. You have not installed the Japanese Language packs on Trading Networks, so Integration Server stores Trading Networks' log entries in U.S. English.

 **Note:** Even if no language packs are installed on the webMethods product and the product is using U.S. English, Integration Server might store log entries from external sources, such as database drivers or adapter resources, in the language used by the operating environment in which the product is running.

2 Setting Up Audit Logging

■ Overview	18
■ Configure Audit Logging	18
■ Set Up Additional Service Logging	24
■ Perform Additional Processing on Audit Log Entries	27

Overview

Integration Server writes error, session, service, security, document, guaranteed delivery, and Mediator transaction audit logging data to files or database tables collectively called the *IS Core Audit Log*. This chapter explains how to set up logging for the IS Core Audit Log.

Integration Server writes business process, integration process, and task audit logging data to database tables collectively called the *Process Audit Log*. For instructions on setting up logging for the Process Audit Log, see *Monitoring BPM, Services, and Documents with BAM: webMethods Monitor User's Guide* (business and integration process audit logging) and *Working with BPM Tasks: webMethods Task Engine User's Guide* (task audit logging).

Configure Audit Logging

Integration Server writes to the IS Core Audit Log using *audit loggers*. Each type of logging data has its own audit logger. For example, the error audit logger writes the audit log entries for errors, the service audit logger writes audit log entries for services, and the document logger writes documents. Each logger has a default configuration, but you can reconfigure it. You do not have to disable a logger to reconfigure it; you can reconfigure an enabled logger.

Start the Logger Configuration

- 1 In Integration Server Administrator, go to the **Settings > Logging** page. The page lists the audit logger for each type of data.



Note: If your Integration Server license does not include security auditing, guaranteed delivery, or Mediator, those loggers are unavailable.

- 2 In the **Logger List**, click a logger you want to set up. The page shows all settings for that logger.
- 3 Click **Edit *type* logger** and set the fields described below.

Enable the Logger

In the **Enabled** area, indicate whether you want the logger to write audit log entries. The default is **Yes** for all loggers except the Security logger.

Choose the Logging Level for the Service Logger

In the **Level** area, choose the level of logging for services. The default is `perSvc`.

Value	Description
<code>perSvc</code>	Lets you set up customized logging on a service-by-service basis in Developer or Designer.
<code>Brief</code>	The logger writes start and failure or start and success log entries for every service every time the service is called, either directly (top-level) or by another service (nested).
<code>Verbose</code>	Same as <code>Brief</code> , except that the logger also writes the input pipeline in all cases.

The `brief` and `verbose` values are globally applied to services; if you choose one of those values, you cannot override it in Developer or Designer for individual services. Software AG recommends using these values only in a development environment, when performing an extensive debugging effort.

Choose the Mode for Writing Log Entries

In the **Mode** area, choose the mode for the logger to use to write log entries. The default is `Asynchronous`.

Value	Description
<code>Synchronous</code>	<p>The logger writes log entries directly to the destination (see “Identify the Destination” on page 21). You might use synchronous logging when you have an application that requires some type of auditing to succeed, and you do not want to proceed without knowing that the auditing occurred.</p> <p>If synchronous logging for an entry fails (for example, because the destination is unavailable), the logger switches to asynchronous mode for that entry.</p>
<code>Asynchronous</code>	The logger writes log entries to a queue, and the queue later writes the entries to the destination. Each logger has its own queue.

Indicate Whether to Persist the Queue

In the **Guaranteed** area, indicate whether Integration Server is to persist the queue on disk when the logger is logging asynchronously. The default is **No**.


Value	Description
No	Integration Server maintains the queue in memory. This option provides better logging performance. However, if Integration Server shuts down abnormally, the log entries in the queue will be lost.
Yes	Integration Server persists the queue on disk. This option is safer but can adversely affect logging performance.

Specify the Maximum Queue Size

For asynchronous logging, in the **Maximum Queue Size** field, specify the maximum number of log entries the logger's queue can hold. The default is 100000. Specify numerals only; for example, do not include commas or periods.

Choose a value that accommodates your system's average volume for log entries. If your logging volume has sudden spikes, the queue can usually catch up by writing the pending entries during lulls. Make sure the Integration Server host machine has enough disk space or memory to accommodate the largest possible size of the queue as specified in this field, as well as the requirements of other applications the Integration Server is hosting. If the logger is writing to a database, the queue's insertion of logged data into the database is constrained by the database's availability and connections limit.

If the queue reaches its maximum, the logger writes the log entries to a file called `FailedAudit_YYYYMMDD_hhmmss.log` in the `Integration Server_directory/logs` directory. You can scan the file to find events that were not logged.

 **Note:** The Service logger cannot write the input pipeline to this file, and the Mediator logger cannot write request and response payloads to this file.


Identify the Destination

In the Destination area, identify where the logger is to write log entries.

Value	Description														
Database	The logger writes log entries to an external RDBMS. You must also set the Maximum Retries and Wait Between Retries fields (see “Specify the Maximum Retries for Database Destination” on page 22 and “Specify the Wait Between Retries for Database Destination” on page 22).														
File	Unless otherwise noted, the logger writes entries to a file in the <i>Integration Server_directory</i> \logs directory, as follows: <table border="1" data-bbox="446 667 1421 1297"> <thead> <tr> <th>Logger</th> <th>Log File Name</th> </tr> </thead> <tbody> <tr> <td>Error</td> <td>WMERROR_YYYYMMDD_hhmmss.log</td> </tr> <tr> <td>Session</td> <td>WMSESSION_YYYYMMDD_hhmmss.log</td> </tr> <tr> <td>Service</td> <td>WMSERVICE_YYYYMMDD_hhmmss.log</td> </tr> <tr> <td>Security</td> <td>WMSECURITY_YYYYMMDD_hhmmss.log</td> </tr> <tr> <td>Guaranteed delivery</td> <td>WMTXIN_YYYYMMDD_hhmmss.log (inbound transactions log file, on the host machine of the Integration Server that is handling guaranteed delivery requests) WMTXOUT_YYYYMMDD_hhmmss.log (outbound transactions file, on the host machine of an Integration Server that is submitting guaranteed delivery requests)</td> </tr> <tr> <td>Mediator transaction</td> <td>MED_EVENT_TXN_YYYYMMDD_hhmmss.log Note: The logger writes the file to the <i>Integration Server_directory</i>/logs/Mediator directory.</td> </tr> </tbody> </table> <p>Note: This value is unavailable for the Document and Process Engine loggers, which can only write to an external RDBMS.</p>	Logger	Log File Name	Error	WMERROR_YYYYMMDD_hhmmss.log	Session	WMSESSION_YYYYMMDD_hhmmss.log	Service	WMSERVICE_YYYYMMDD_hhmmss.log	Security	WMSECURITY_YYYYMMDD_hhmmss.log	Guaranteed delivery	WMTXIN_YYYYMMDD_hhmmss.log (inbound transactions log file, on the host machine of the Integration Server that is handling guaranteed delivery requests) WMTXOUT_YYYYMMDD_hhmmss.log (outbound transactions file, on the host machine of an Integration Server that is submitting guaranteed delivery requests)	Mediator transaction	MED_EVENT_TXN_YYYYMMDD_hhmmss.log Note: The logger writes the file to the <i>Integration Server_directory</i> /logs/Mediator directory.
Logger	Log File Name														
Error	WMERROR_YYYYMMDD_hhmmss.log														
Session	WMSESSION_YYYYMMDD_hhmmss.log														
Service	WMSERVICE_YYYYMMDD_hhmmss.log														
Security	WMSECURITY_YYYYMMDD_hhmmss.log														
Guaranteed delivery	WMTXIN_YYYYMMDD_hhmmss.log (inbound transactions log file, on the host machine of the Integration Server that is handling guaranteed delivery requests) WMTXOUT_YYYYMMDD_hhmmss.log (outbound transactions file, on the host machine of an Integration Server that is submitting guaranteed delivery requests)														
Mediator transaction	MED_EVENT_TXN_YYYYMMDD_hhmmss.log Note: The logger writes the file to the <i>Integration Server_directory</i> /logs/Mediator directory.														

The first time you start Integration Server, the settings depend on whether the IS Core Audit Log function in Integration Server connects to an external RDBMS, as follows:


- If it does not connect, the Error, Session, Service, Security, Guaranteed Delivery and Mediator transaction loggers are set to **File**. (The Document and Process Engine loggers can only write to RDBMS and so are unavailable.)
- If it connects, all loggers are set to **Database**. However, if the IS Core Audit Log function cannot connect when you start Integration Server for a subsequent session (for example, because the RDBMS is unavailable), the loggers are reset to **File**, and you will have to set them to **Database** again yourself.

 **Note:** The IS Core Audit Log was configured during or after Integration Server installation. For Mediator transaction events, the Mediator Events database component was installed. For details, see the *Software AG Installation Guide*.

Specify the Maximum Retries for Database Destination

If the destination is set to **Database**, in the **Maximum Retries** field, specify the number of times the logger (synchronous) or queue (asynchronous) should try to connect to the database. The default is 3 retries.

- For synchronous logging, the logger tries to write each log entry up to the number of times specified, then switches to asynchronous logging.
- For asynchronous logging, the queue tries to write each log entry up to the number of times specified, then writes the entry to a file called `FailedAudit_yyyymmdd_hhmmss.log` in the *Integration Server_directory/logs* directory. You can scan the file to find events that were not logged.

 **Note:** The Service logger cannot write the input pipeline to this file, and the Mediator logger cannot write request and response payloads to this file.

Specify the Wait Between Retries for Database Destination

If the destination is set to **Database**, in the **Wait Between Retries** field, specify the number of seconds the logger should wait between tries to connect to the RDBMS. The default is 5 seconds.

Additional Fields for the Security Logger

If you are configuring the Security logger, set the additional fields below.

Choose Whether to Generate Auditing Data on Startup

By default, the security logger writes security events that occur after Integration Server has completed its startup sequence. In the **Generate Audit Data on Startup** area, choose whether the logger should also write security events that occur during Integration Server's startup sequence. The default is **No**.

 **Note:** Writing security events during startup makes the startup sequence significantly slower.

Choose When to Generate Security Auditing Data

In the **Generate Auditing Data on** area, choose when Integration Server should log security events. The default is **Success or Failure**.

Value	Description
Success	Only when the event completes successfully.
Failure	Only when the event fails.
Success or Failure	Regardless of outcome.

Choose Security Areas to Audit

In the **Security Areas to Audit** area, select the areas for which to log security-related events.

Complete the Logger Configuration

When you are done settings the fields, click **Save Changes** and then restart Integration Server.

Set Up Additional Service Logging


If you selected `perSvc` logging for the Services logger, you must set up customized logging in Developer or Designer for every service you want to audit.

You can augment any type of service logging by using Integration Server built-in services to write user-defined messages to the Integration Server server log or the IS Core Audit Log.

Set Up Customized Service Logging in Developer or Designer

For each service, you can choose the following logging options. For complete information on working with services, see *Developing Integration Solutions: webMethods Developer User's Guide* or Designer online help.


- Whether to log and, if so, when, as follows:
 - Every time the service is called, whether by a client request, trigger, or another service.
 - Only when the service is called by a client request or a trigger (that is, when the service is a top-level service).
- The statuses in the service's execution on which to log - when the service fails, fails or succeeds, or starts and fails or succeeds.
- Whether to store the service's input pipeline and, if so, always or only when an error occurs. Storing the input pipeline allows you to resubmit the service later if necessary.

 **Note:** You can only log input pipelines if you are writing service data to an external RDBMS.

To improve service logging performance, do the following:

- Set up customized logging for top-level services only. Avoid logging nested services.
- Log on service failure or log on service failure or success. Only choose to log on service failure or success *and* start when you need the greatest possible quality of service.
- Logging the pipeline can negatively affect performance, especially if the pipeline contains large objects, because Integration Server has to make a copy of the pipeline every time the service is invoked. Store the input pipeline only for top-level services, and only when absolutely necessary (for example, on failure only). Remove all unnecessary data from the pipeline to minimize the volume of data to store.
- The audit log entries that the Process Engine can write for business process steps that run services convey the same information as the audit log entries you can write for services. In addition, the Process Engine can store the input pipeline for services that

are run by process steps. To improve logging performance, avoid logging the same information twice by coordinating audit logging for services that are invoked by process steps.

 **Note:** When coordinating logging, keep in mind that when a service is run by a process step, it is actually called by a wrapper service, making it a nested service rather than a top-level service.

Write User-Defined Messages or Input Pipelines to the Integration Server Server Log

You can have running services post user-defined progress messages to the Integration Server server log. For example, you might have a service post messages to indicate that certain pieces of code ran successfully, or to record run-time values for variables so you can see how the values changed as the service ran. To do so, you make the service call the Integration Server built-in service `pub.flow:debugLog`.

You can also have running services write input pipelines to the Integration Server server log. To do so, you make the service call the Integration Server built-in service `pub.flow:tracePipeline`.

You can write this information regardless of how you have configured service audit logging. For instructions on using these services, see the *webMethods Integration Server Built-In Services Reference* and *Developing Integration Solutions: webMethods Developer User's Guide*. For information on the Integration Server server log, see *Administering webMethods Integration Server*.

Write User-Defined Messages to the IS Core Audit Log

If you are storing service audit data in an external RDBMS, and you have installed the Process Engine, you can have services post user-defined progress messages to the IS Core Audit Log. For example, you might have a service post messages to indicate that certain pieces of code ran successfully, or to record run-time values for variables so you can see how the values changed as the service ran. To do so, you make the service call the Integration Server built-in service `pub.prt.log:logActivityMessages`.

 **Note:** You view these messages in Monitor.

The `pub.prt.log:logActivityMessages` service is stored in the WmPRT package. Its input and output parameters are described below.

Input Parameters

<i>FullMessage</i>	String Optional. Complete message to record in the IS Core Audit Log. The message can be up to 1024 bytes.
<i>BriefMessage</i>	String Optional. Shortened version of the full message. The message can be up to 240 bytes.

<i>EntryType</i>	String Type of message.
<u>Set to...</u>	<u>To indicate that the message is...</u>
Message	Informational. No action is needed.
Warning	A warning message. The service can complete successfully even if the circumstance causing the warning is not addressed.
Error	An error message. An error message will not stop the service or put it in an error state. However, the service cannot complete successfully until the circumstance causing the error is resolved. This is the default.

Output Parameters

None.

Send Messages About Service Failures to Email Addresses

You can configure Integration Server to automatically send notifications to a specified e-mail address each time a service fails. These service failures are the stack track data written to the error log. In a development environment, you might direct these messages to the developer. In a production environment, you might direct them to the Integration Server administrator.

► To send messages about service failures to e-mail addresses

- 1 In Integration Server Administrator, go to the **Settings > Resources** page and click **Edit Resource Settings**.
- 2 In the **SMTP Server** field, type the server name or IP address of the SMTP server to use to send the messages.
- 3 In the **Internal Email** field, type the e-mail address to which to send messages about critical log entries. Typically, you would specify the email address for the Integration Server administrator.
- 4 In the **Service Email** field, type the e-mail address to which to send messages about service failures. In a development environment, you might direct these messages to the developer. In a production environment, you might direct these messages to the Integration Server administrator.
- 5 By default, Integration Server uses character set UTF-8 for the messages. If you want to use a different character set, identify the character set in the **Default Email Charset** field.
- 6 Click **Save Changes**.

- 7 By default, Integration Server connects to port 25 on the specified SMTP server. Also by default, when sending a message, Integration Server provides its own address (the From Address) as `Integration-Server@localhost`, where *localhost* is the Integration Server host machine. If you want to change either of these properties, follow these steps:
- a In Integration Server Administrator, go to the **Settings > Extended** page. Integration Server Administrator displays a list of Integration Server configuration properties you can change using this method.
 - b Click **Edit Extended Settings**. In the **Extended Settings** box, set the properties as follows:

To change this...	Set this...
SMTP server port	<code>watt.server.smtpServerPort=port to use</code>
Integration Server's From Address	<code>watt.server.email.from=new From Address to use</code>

- c Click **Save Changes**, then restart Integration Server.

Perform Additional Processing on Audit Log Entries

If you want to perform additional processing on log entries, you can create an event handler. For example, you could create an event handler that sends service log entries to another log, such as the Event Log on a Windows system. For information, see the *webMethods Integration Server Built-In Services Reference* and *Developing Integration Solutions: webMethods Developer User's Guide*.

3 Viewing Audit Log Data

■ Overview	30
■ View the Audit Logs in Integration Server Administrator	31
■ View the Mediator Transaction Logs	34
■ Change the Log Displays	36

Overview

You can use Integration Server Administrator, Monitor, or both to view audit log data.

Audit Log Data	View using Integration Server Administrator?	View using Monitor?
Documents	No	✓Yes
Errors	✓Yes	✓Errors for logged services, documents, and processes
Guaranteed delivery	✓Yes	No
Security	✓Yes	No
Services	✓All except logged input pipelines and user-defined messages in the IS Core Audit Log	✓Yes
Sessions	✓Yes	No
Business processes	No	✓Yes
Tasks*	No	No
Integration processes	No	✓Yes
Mediator transaction**	No	No

* For information on viewing logged data for tasks, see *Working with BPM Tasks: webMethods Task Engine User's Guide*.

** To view logged data for Mediator transactions, you must open the log file manually or look up the data in the MED_EVENT_TXN table. For more information, see ["View the Mediator Transaction Logs"](#) on page 34.

Monitor links related logged data in its display; for example, for a business process or business process step, you can see all relevant service, error, and user-defined message entries. You can also perform a variety of actions from Monitor; for example, if you logged input pipelines for services, you can edit the pipelines and resubmit the services, and you can archive or delete audit log data. For complete information, see *Monitoring BPM, Services, and Documents with BAM: webMethods Monitor User's Guide*.

Integration Server Administrator does not link related data for you. You must look through the individual logs for related data yourself. This chapter explains how to view audit logs in Integration Server Administrator and change various aspects of the log displays.

View the Audit Logs in Integration Server Administrator

By default, Integration Server Administrator displays the most recent entries in the logs.

View the Error Log

In Integration Server Administrator, go to the **Logs > Error** page to view the error log. The fields in the error log are listed below.

Column	Details
Timestamp	Date and time the entry was written to the log.
Service Name	Name of the service in which the error occurred.
Service Stack	Parent services for the service in which the error occurred.
Error Message	Message that describes the error that occurred.
Stack Trace	Trace that shows the call sequence leading to the error. To expand the display of stack trace data, select the Expand Stack Trace Data check box in the Log display controls area and click Refresh .
Root Context Parent Context Current Context	Context information Monitor uses to connect related entries from different logs.

View the Guaranteed Delivery Log

In Integration Server Administrator, go to the **Logs > Guaranteed Delivery** page to view the guaranteed delivery log. The fields in the guaranteed delivery log are listed below.

Column	Details
Timestamp	Date and time the entry was written to the log.
Status	Current status of the transaction (Start or Stop).
Message	Name of the guaranteed delivery process that is running.
Error Message	If the transaction failed, message that describes the error that occurred.
Root Context Parent Context Current Context	Context information Monitor uses to connect related entries from different logs.

Integration Server writes guaranteed delivery log entries to two logs, one for inbound transactions and one for outbound transactions. By default, Integration Server Administrator displays the most recent entries in the inbound guaranteed delivery transactions log. You can switch to the log entries in the outbound transactions log by clicking **View Guaranteed Delivery Outbound Log**.

View the Security Log

In Integration Server Administrator, go to the **Logs > Security** page to view the security log. The fields in the security log are listed below.

Column	Details
Timestamp	Date and time the entry was written to the log.
Message	Text that explains the security event that occurred.
Server Id	Integration Server on which the security event occurred. This is necessary information when Integration Servers are clustered and writing to a shared RDBMS. The ID can be <i>DNSname:port</i> or <i>IPaddress:port</i> . Note: The <i>port</i> is always the Integration Server's primary port, even if the event occurred on a different (non-primary) Integration Server port.
Client Id	Network IP address for the client from which the security event was performed.
User Id	Integration Server user name under which the client connected to perform the security event.
Security Event Type	Category for the security event that occurred (authentication, authorization, certificates, configuration, and so on).

View the Service Log

In Integration Server Administrator, go to the **Logs > Service** page to view the service log. The fields in the service log are listed below.

Column	Details
Timestamp	Date and time the entry was written to the log.
User Id	Integration Server user name of the client that called the service that generated the log entry.
Server Id	Integration Server on which the service that generated the log entry ran. This is necessary information when Integration Servers are clustered and writing to a shared RDBMS. The ID can be <i>DNSname:port</i> or <i>IPaddress:port</i> . Note: The <i>port</i> is always the Integration Server's primary port, even if a service executed on a different (non-primary) Integration Server port.
Service Name	Service that generated the log entry.

Column	Details
Resubmittable	Whether you can resubmit the service from Monitor. You can resubmit a service if it is a top-level (as opposed to nested) service and the service's input pipeline was logged.
Status	Current status of the service (Started , Retried , Ended , or Failed).
Duration	Length of time the service ran (in milliseconds).
Error Message	If the service failed, message that describes the error that occurred.
Root Context Parent Context Current Context	Context information Monitor uses to connect related entries from different logs.

For information about viewing service log entries in Monitor, see *Monitoring BPM, Services, and Documents with BAM: webMethods Monitor User's Guide*.

View the Session Log

In Integration Server Administrator, go to the **Logs > Session** page to view the session log. The fields in the session log are listed below.

Column	Details
Timestamp	Date and time the entry was written to the log.
Server Id	Integration Server on which the session occurred. This is necessary information when Integration Servers are clustered and writing to a shared RDBMS. The ID can be <i>DNSname:port</i> or <i>IPaddress:port</i> . Note: The field lists the Integration Server's primary port, even if the session occurred on a different (non-primary) Integration Server port.
User Id	Integration Server user name under which the client connected for the session.
Client IP	IP address of the machine from which the client request was submitted. The word "system" appears for session requests from Integration Server for operations such as running a scheduled service or refreshing the display.
Session State	Current status of the session (Started , Expired , or Ended).
RPCs	Number of services the client has called so far during the session.
Age	Duration of the session, in milliseconds.
Session ID	A string the server generates to uniquely identify each session.

View the Mediator Transaction Logs

You cannot view logs for Mediator transactions using the Integration Server Administrator or Monitor. You can view logs for Mediator transactions only by manually opening either:

- **The log file (the flat file).** You can open the log file using a text editor. The log file is located in the *Integration Server_directory/logs/Mediator* directory.
- **The audit table.** You can open the audit table using your RDBMS editor. The table name is `MED_EVENT_TXN`. For more information, see the documentation for your RDBMS editor.

The columns in the Mediator transaction log are listed below.

Flat File Column Name	Database Column Name	Details
Session ID	SESSION_ID	SOAP invocation session.
Service Name	SERVICE_NAME	Name of the virtual service that generated the log entry.
Target Name	TARGET_NAME	Name of the Mediator instance reporting the event.
Consumer	CONSUMER_NAME	Service consumer name associated with the call. This is included when an Identify Consumer policy action is defined for the virtual service.
Consumer IP	CONSUMER_IP	IP address of the service consumer. This is included when an Identify Consumer policy action is defined for the virtual service.
Request Status	STATUS	Current status of the request (Success or Failure).
Response Payload	RESPONSE	Response payload. This field is written only if you use the Mediator database component. It cannot be written to the Mediator log file.
Request Payload	REQUEST	Request payload. This field is written only if you use the Mediator database component. It cannot be written to the Mediator log file.
Total Roundtrip Time	TOTAL_TIME	Time in milliseconds required to invoke the service provider. This time includes the overhead incurred by Mediator. Overhead includes security overhead for encryption, decryption, and load-balance retries.


Flat File Column Name	Database Column Name	Details
Provider Roundtrip Time	PROVIDER_TIME	Time in milliseconds required for Mediator to invoke a service provider and receive a response.
Insert Timestamp	INSERTTIMESTAMP	Date and time the entry was written to the log. This is calculated by the RDBMS.
Timestamp	AUDITTIMESTAMP	Date and time the audit entry was created. This is calculated by the Integration Server.
Root Context Id	ROOTCONTEXTID	Globally unique identifier (GUID) for the Mediator transaction event.
Parent Context Id	PARENTCONTEXTID	
Context Id	CONTEXTID	
Message Id	MSGID	GUID used as the primary key for the row.
Server Id	SERVERID	<p>ID of the host machine that produced the audit record. The ID can be <i>DNSname:port</i> or <i>IPaddress:port</i>.</p> <p>Note: The <i>port</i> is always the Integration Server's primary port, even if the event occurred on a different (non-primary) Integration Server port.</p>

Change the Log Displays

You can change the display of log pages in Integration Server Administrator. You can:

- Display logged data in different languages.
- Change various aspects of the display for all logs permanently.
- Change various aspects of a particular log's display temporarily.

Display Logged Data in Different Languages

 **Note:** The changes in this section will also affect the Integration Server server log, described in *Administering webMethods Integration Server*.

This section applies only to logged data that is stored in files.


If you want to view logged data in a language other than English, you might have to adjust your text editor or command shell. Integration Server writes the files in the Unicode UTF-8 encoding. These files do not contain a Byte Order Mark (BOM, Unicode character U+FEFF). If the files contain non-ASCII data, such as log entries written in non-U.S. English, you might have to adjust the character encoding used by your text editor or command shell so you can view the log entries.

On a UNIX system, you can adjust the character encoding by changing your locale setting (LC_ALL) to the appropriate UTF-8 encoded locales. For example, to view Japanese characters in a text editor or command shell on a Solaris system, you might change your locale setting to `ja_JP.UTF-8`.

On a Windows system, because the files do not contain the BOM character, text editors such as Notepad might not detect the UTF-8 encoding correctly. Adjust the encoding manually so you can view the files. To view the logs in the cmd shell, you can use the command `chcp 65001`.

Change the Display Permanently for All Logs

By default, the number of log entries shown for logs in Integration Server Administrator is 35 and the refresh interval is 90 seconds. You can change these defaults.

 **Important!** Significantly increasing the number of entries displayed or decreasing the refresh interval can slow system performance. Changing these properties will also affect the Integration Server server log, described in *Administering webMethods Integration Server*.

Also by default, the timestamps in the log entries default to local time and display the time zone. You can change this to the Coordinated Universal Time (UTC) that is recorded for the entries in the IS Core Audit Log database component.

- 1 In Integration Server Administrator, go to the **Settings > Extended** page and click **Show and Hide Keys**. Integration Server Administrator displays a list of the Integration Server configuration properties you can change using this method.
- 2 Select the check box next to each property you want to change, as follows:

If you want to change...	Select this property...
Number of log entries shown	watt.server.log.maxEntries
Refresh interval for log display	watt.server.log.refreshInterval
Timestamp for log entries to UTC	watt.server.audit.displayLogs.convertTime


- 3 Click **Save Changes**. Integration Server Administrator displays the selected properties in the **Extended Settings** box.
- 4 Click **Edit Extended Settings**. In the **Extended Settings** box, set the properties as follows:

To change...	Property	Set to...
Number of log entries shown	watt.server.log.maxEntries	Positive integer
Refresh interval for log display	watt.server.log.refreshInterval	Positive integer
Timestamp for log entries to UTC	watt.server.audit.displayLogs.convertTime	false

- 5 Click **Save Changes**. Changes take effect immediately.

Change the Display Temporarily for a Particular Log

To change the display for a particular log temporarily, use the **Log display controls** area at the top of the log display page and then click **Refresh**. The changes remain until you change them again, or until you shut down Integration Server, whichever comes first.

 **Note:** If Integration Server is storing logged data in an external RDBMS, most log pages offer **From:** and **To:** fields that let you choose the entries to display using a date range. However, using date ranges can slow system performance.