Infrastructure Series

TechDoc
WebSphere Message Broker / IBM Integration Bus

Using Logs
(Administration)

March 2015 - IBM Message Broker - Administration – Using Logs
Author(s): Glen Brumbaugh

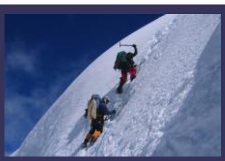

Table of Contents

Introduction 3

Document Version 3

Product Naming History 3

Product Component Terminology 3

Using Logs 4

Logging Overview 4

Standard System Logs (stdout, stderr) 4

Local Error Logs (syslog) 5

Windows Local Error Logs (Events) 5

UNIX Local Error Logs (syslog) 5

Log Filters 5

Message Broker Toolkit Logs 6

Integration Explorer (MQExplorer) Logs 6

UNIX syslog Configuration 6

syslog daemon 6

rsyslog daemon 7

syslog-ng daemon 8

Best Practices 9

References 9

Introduction

Document Version

This document describes how to configure and use the logging information available in WebSphere Message Broker (WMB up to v8.x) or IBM Integration Bus (IIB v9.0+). The document should, however, apply to most versions of these products. The contents of this document have been specifically verified on the following production versions:

WebSphere Message Broker	v7.0.0.2
IBM Integration Bus	v9.0.0.0

This documentation has been created and maintained by:

- Glen Brumbaugh

This documentation has been reviewed by:

- Glen Brumbaugh

This documentation was last updated:

- Version 1.0 March 2015

Product Naming History

The product currently known as IBM Integration Bus has been through a number of different product names during its several decade long evolution. The product was originally developed by the New Era of Networks (NEON) Corporation and was marketed and resold by IBM. IBM completely redesigned and rebuilt the product and released their own in-house developed product beginning with version 2.0. The product has had the following names and version numbers:

MQSeries Integrator (MQSI)	Version 1.0 – 2.0
WebSphere MQSeries Integrator	Version 2.1
WebSphere Business Integration Message Broker (WBIMB)	V5.0
WebSphere Message Broker (WMB)	Version 6.0 - 8.x
IBM Integration Bus (IIB)	Version 9.0 - 10.0

For the remainder of this document, the product will be referred to as "*Message Broker*". This is both for historical reasons and to signify that this documentation applies to both the WMB and IIB product versions.

Product Component Terminology

With the Version 9.0 product rename (to IBM Integration Bus), several key product architectural components were given new names; while continuing to fill virtually the same role they had previously filled. This documentation will continue to refer to the "old" names because the information documented here refers to both the old and new product versions.

The old and corresponding new names are as follows:

- Message Broker → Now called "Integration Node" (Beginning with v9.0)
- Execution Group → Now called "Integration Server" (Beginning with v9.0)
- Message Flow → Still called "Message Flow"

Using Logs

Logging Overview

The Message Broker uses a number of different logs to record information. These logs include:

- Standard System logs (stdout, stderr)
- Local Error logs (syslog)
- Message Broker Toolkit logs

In general, problem determination should begin with these logs and they should be searched in the order listed. In addition to these Message Broker logs, information related to Message Broker processing may also be found in other software product logs. These logs include:

- HTTP Server logs (Apache, IBM IHS, etc.)
- Web Server logs (Apache Tomcat, WebSphere, etc.)
- WebSphere MQ logs
- Database logs (DB2, Oracle, etc.)

Standard System Logs (stdout, stderr)

Standard System logs are supported on all distributed platforms, although the implementation details (number of files, names, etc.) may vary by platform. The two logs supported are:

- Standard Output (stdout)
- Standard Error (stderr)

Note that log messages may be generated either by a "Broker" or by an "Execution Group". There are different log locations for each of these components. Again, these details vary by platform. These log details are as follows:

UNIX/Linux

• Message Broker	/var/mqsi/components/*BrokerName*/stdout /var/mqsi/components/*BrokerName*/stderr
• Execution Group	/var/mqsi/components/*BrokerName*/*ExecutionGroupID*/stdout /var/mqsi/components/*BrokerName*/*ExecutionGroupID*/stderr

Windows

• Message Broker	/*WorkPath*/components/*BrokerName*/console.txt
• Execution Group	/*WorkPath*/components/*BrokerName*/*ExecGroupID*/console.txt

Note that Message Broker logs, but not Execution Group logs, may be viewed through the MQExplorer tool if the Message Broker is both running and connected. Within MQExplorer, these logs are titled "Administration Log". This assumes, of course, that the IBM Integration Explorer plug-in has been installed into the MQExplorer tool.

Note: Both the Message Broker and Execution Group logs will normally only contain start times for their respective components. Any other information in these logs may need to be reviewed.

Local Error Logs (syslog)

Windows Local Error Logs (Events)

In addition to writing routine operational messages to the standard system logs (stdout, stderr), Message Broker components also write messages to the Operating System (OS) log (syslog). While all distributed Operating Systems provide OS logging, the implementation mechanism of these logs again varies by platform. On Windows servers, these log messages are store in Windows Event log under the "Application" view. These messages can be viewed using the following Windows tool:

- Event Viewer (eventvwr.exe) Windows Logs / Application
    - Find "IBM Integration" (IIB v9.0)
      "WebSphere Broker" (WMB v8.0)
      "WebSphere Broker" (WMB v7.0)

UNIX Local Error Logs (syslog)

The OS syslog must be configured to process these Message Broker log entries. Since Message Broker relies upon the underlying OS capabilities, the specific OS syslog configuration details may vary both by Operating System and vendor. On UNIX, there are three different syslog daemons in common use. These are:

- syslogd (Oldest daemon; Limited to 12 real and 8 local facilities)
- syslog-ng (Default in SuSE Linux Enterprise Server - SLES)
- rsyslogd (Default in Red Hat Enterprise Linux - RHEL)

The various implementations all derive from a common base and share some architectural properties. All messages are written to a named "Facility". The "Facility" is a container to separate messages from different sources. The daemons vary in the number of predefined and optional "Facilities" that they can handle. Each log message written to a "Facility" also has a "Log Level". This level separates messages by severity. Again, the daemons vary in the number and type of their "Log Levels".

The Message Broker generated log entries have the following characteristics:

- Facility: User (Facility available on all daemons)
- Log Levels: Info, Warning, Err (Log Levels available on all daemons)

As can be seen, Message Broker only uses three (3) of the available Log Levels. This was done to ensure compatibility across various syslog implementations. All of the Log Levels in a syslog have an associated numerical priority in addition to their name. Filters can be constructed to select messages of an importance "greater than or equal to" the indicated level. The Message Broker Log levels have the following priorities across all syslog platforms:

- Err (Highest level of priority)
- Warning (Middle level of priority)
- Info (Lowest level of priority)

Log Filters

System logs typically have "filters" that determine which messages sent to the log are retained and/or displayed. The default behavior is normally to discard all messages unless a "filter" exists to select that message. The function of filters is to be "Selectors" and to select messages for inclusion in the syslog, not to filter messages for exclusion. The mechanism for specifying filters

naturally varies by the daemon in use. These specification mechanisms are described in later Sections.

Message Broker Toolkit Logs

There are a number of logs in the Message Broker Toolkit that may be helpful in non-production environments. These logs include:

• Eclipse Error Log	(Internal error from Eclipse & Message Flows)
• Toolkit Deployment Log	(Message Flow and object deployment information)
• Eclipse Problems View	(Message Flow and resource errors)
• Tag Delimited String (TDS) log	(TDS format description errors)

Message Broker Toolkit logs can be found in the following locations (on Windows):

• Deployment log	(*Drive*:\\Users\UserID\IBM\TookitVersion\Workspace\\.metdata\plugins \com.ibm.etools.mft.broker.runtime)
• Eclipse Error log	(*Drive*:\\Users\UserID\IBM\TookitVersion\Workspace\\.metdata\\.log)
• Eclipse TDS log	(*Drive*:\\InstallDir\log\TDS.log)

Integration Explorer (MQExplorer) Logs

There is a very useful log that is only available through the Integration Explorer (MQExplorer with Broker plug-in). This log is the:

- Activity log

Activity logs provide a high-level view of how Message Flows interact with other resources. They are thus extremely useful in the development and testing environments. These logs are related either to a specific Message Flow or a specific Resource Type. These logs are accessed through MQExplorer as follows:

- MQExplorer Integration Nodes / *Broker* / *ExecGroup* / *MsgFlow* /
  Right click and select "Open Activity Log"

- MQExplorer Integration Nodes / *Broker* / *ExecGroup* / Resource Managers/*Resource*
  Right click and select "Open Activity Log"

UNIX syslog Configuration

syslog daemon

The syslog daemon uses the following configuration file:

- /etc/syslog.conf

The syslog daemon uses the following configuration syntax:

- *Facility*.*Level* *Destination*

The following notes apply to syslog daemon configuration lines:

• *Facility*	Must be "User"	
• *Level*	"*"	→ Select all messages
	"Info"	→ Select "Info", "Warning" and "Err" messages
	"Warning"	→ Select "Warning" and "Err" messages
	"Err"	→ Select "Err" messages

- There must be a "tab" and not a "space" between the Level and the Destination!
- *Destination* *Path*/*File* → Send messages to a file (Path must begin with "/").
  UserID → Send messages to the specified User ID via sendmail.
  @*Hostname* → Send messages to a remote syslog.

Note that the destination specified will receive all "User" facility messages, not just the Message Broker messages! The syslog daemon was originally developed to support sendmail and is somewhat restrictive. This is the reason for the other syslog implementations. Messages can be selected using a command like "sed" to select messages using the same Selection criteria as was described for Windows Events.

rsyslog daemon

By default, the rsyslog daemon uses the following configuration file:

- /etc/rsyslog.conf

Note that this configuration file can be overridden by using the "-f" parameter when launching rsyslog. The rsyslog daemon has sophisticated configuration file syntax. The complete particulars of this syntax are beyond the scope of this document. A brief summary of its major capabilities is, however, provided here.

The rsyslog daemon supports three different styles of filters. These three styles are:

- Syslog style Selectors (facility.level destination)
- Property Based Filters
- RainerScript

Property Based Filter Syntax

There are a considerable number of properties that may be used to select messages in Property Based filter syntax. For a complete list of properties, refer to the "rsyslog" site in the "References" Section. The most relevant properties for Message Broker are the following:

• msg	(The message)
• programname	(Program name for the message tag)
• syslogfacility-text	(Should be "User").
• Syslogseverity-text	(Should be "Info", "Warning", or "Err")

The syntax for Property Based filters is as follows"

- :*Property*, [!]*Comparison*, "*Value*"

The following notes apply to rsyslog Property Based configuration lines:

- ":" must be in the first column of the record.
- *Property* May be any rsyslog supported property.
- *Comparison* "contains" → Message contains the specified string.
  "isequal" → Message equals the specified string.
  "startswith" → Message starts with the specified string.
  "regex" → Compares using the specified regular expression.
  "ereregex" → Compares using the extended regular expression.
- *Value* Value must be quoted. "Property" compared to "Value".
- The "Bang" ("!") symbol negates the comparison.

Obviously, Property Based filters also require a “destination”. In rsyslog, “destinations” are described as “Actions”. Action syntax is common across all three types of filter styles. Again, see the rsyslog site in the “References” Section for more information.

Note: The rsyslog organization no longer recommends the usage of property-based filters!

RainerScript Filter Syntax

RainerScript filters are based upon “IF-THEN-ELSE” blocks. These blocks can be nested and arbitrarily complex comparisons can be created. Comparisons can include Boolean, Arithmetic, and String operations. “IF” and “ELSE” blocks are multi-statement blocks, thus allowing further levels of nesting as well as complex operations to be performed.

A variety of data types, operators, and comparisons are supported. These include:

- Parenthesis
- Not, unary minus
- *, /, and % (Modulus; like “C”)
- +, - , & (Concatenation)
- ==, !=, <>, <, >, <=, =>, contains, startswith,
- contains_i, startswith_i (Both ignore case)
- and, or
- built-in functions

A sample RainerScript filter might be as follows:

```
If    ($msg contains “WebSphere Broker”)
or    ($msg contains “IBM Integration”)  then
{
      /var/log/mqsi.log
}
```

Note: RainerScript are the rsyslog organization recommended syntax!

syslog-ng daemon

By default, the syslog-ng daemon uses the following configuration file:

- /etc/syslog-ng/syslog-ng.conf

The syslog-ng configuration file contains a sophisticated syntax and a complete description is beyond the scope of this document. For additional information, see the “syslog-ng” site in the “References” Section. The syslog-ng configuration file contains the following types of statements:

Source	(Log message sources – Input)
Destination	(Log message destinations - Output)
Log	(Connect “Source” to “Destination” using an optional “Filter”)
Filter	(Identify whether a “Source” message should be processed)
Parser	(Define parsing of input log messages into fields for better filters)
Rewrite	(Modify input messages for the output destination)
Template	(Used for standardization; messages, names, etc.)

These statements have the following standard syntax:

- *Type   Name*   {*Option1(parm1a, parm1b); Option2(parm2a, parm2b); …*};

- The Parameters of an object are enclosed in braces (“{“, “}”).
- The statements end with a semicolon (“;”).
- An “Options List” is a semicolon (“;”) separated list.
- Each Option in the list has a name and a comma (“,”) separated list of parameters. This Option parameter list is enclosed in parenthesis “(“, “)”.
- Option parameters are both positional and optional.
- Optional parameters have the format of: *parmName*(*Value*).
- The configuration file is case sensitive.

The syslog-ng daemon supports a significant number of different “Sources” and “Destinations”

A sample syslog-ng filter might be as follows:

```
# Selector for Message Broker log messages #

source internalLog { unix-stream(“/dev/log”, group(log), max-connections(10)); };

destination ibmbroker { file("/var/log/mqsi.log" perm(0644) ); };

filter f_ibmbroker1  { match('WebSphere Broker'); };
filter f_ibmbroker2  { match('IBM Integration’); };

log { source(internalLog); filter f_ibmbroker1); destination(ibmbroker); };
log { source(internalLog); filter f_ibmbroker2); destination(ibmbroker); };
```

Best Practices

- **Create an OS Filter for syslog Messages:** The Message Broker log messages will be discarded unless there is a filter entry for the System log capturing the messages.

- **Write the Message Broker System Log messages to a dedicated Message Broker file:** There may be a high volume of log file messages and these messages will normally only be of interest to the Message Broker administrators and developers, not to the OS System Administrators.

References

- ○ IBM – Knowledge Center – IIB – v9.0 – Logs
  http://www-01.ibm.com/support/knowledgecenter/?lang=en#!/SSMKHH_9.0.0/com.ibm.etools.mft.doc/au14160_.htm

- ○ IBM – Knowledge Center – IIB – v9.0 – Using Logs
  http://www-01.ibm.com/support/knowledgecenter/?lang=en#!/SSMKHH_9.0.0/com.ibm.etools.mft.doc/an04230_.htm

- IBM – Knowledge Center – IIB – v9.0 – Standard System Logs (stdout, stderr)
  http://www-01.ibm.com/support/knowledgecenter/?lang=en#!/SSMKHH_9.0.0/com.ibm.etools.mft.doc/au14165_.htm

- IBM – Knowledge Center – IIB – v9.0 – Local Error Logs (syslog)
  http://www-01.ibm.com/support/knowledgecenter/#!/SSMKHH_9.0.0/com.ibm.etools.mft.doc/an01300_.htm

- IBM – Knowledge Center – IIB – v9.0 – Installing IBM Integration Explorer
  http://www-01.ibm.com/support/knowledgecenter/?lang=en#!/SSMKHH_9.0.0/com.ibm.etools.mft.doc/ah38100_.htm

- Syslog-ng – Organization – Home Page
  https://syslog-ng.org

- Balabit – syslog-ng – Home Page
  http://www.balabit.com/support/documentation/

- Rocket-Fast System for Log Processing – rsyslog – Home Page
  http://www.rsyslog.com/doc/rsyslog_conf_filter.html